

Universidade Federal de Ouro Preto – UFOP
Instituto de Ciências Sociais Aplicadas
Graduação em Economia

Aplicações dos Algoritmos Genéticos ao Modelo Oligopolista de Cournot

Izabelle Angel Martins Heredia

Mariana, Minas Gerais

2018

Izabelle Angel Martins Heredia

Aplicações dos Algoritmos Genéticos ao Modelo Oligopolista de Cournot

Monografia apresentada ao Curso de Ciências Econômicas do Instituto de Ciências Sociais Aplicadas (ICSA) da Universidade Federal de Ouro Preto (UFOP), como requisito à obtenção de grau de Bacharel em Ciências Econômicas.

Universidade Federal de Ouro Preto – UFOP

Instituto de Ciências Sociais Aplicadas

Graduação em Economia

Orientador: Prof. Dr. Martin Harry Vargas Barrenechea

Mariana, Minas Gerais

2018

H542a Heredia, Isabelle Angel Martins.
Aplicações dos algoritmos genéticos ao modelo oligopolista de Cournot
[manuscrito] / Isabelle Angel Martins Heredia. - 2018.

58f.: il.: grafs; tabs.

Orientador: Prof. Dr. Martin Harry Vargas Barrenechea.

Monografia (Graduação). Universidade Federal de Ouro Preto. Instituto de
Ciências Sociais Aplicadas. Departamento de Ciências Econômicas e
Gerenciais.

1. Algoritmo genético - Teses. 2. Economia - Teses. I. Barrenechea, Martin
Harry Vargas. II. Universidade Federal de Ouro Preto. III. Título.

CDU: 519.8

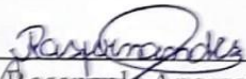
IZABELLE ANGEL MARTINS HEREDIA

Curso de Ciências Econômicas - UFOP

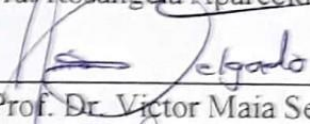
APLICAÇÃO DOS ALGORITMOS GENÉTICOS AO MODELO OLIGOPOLISTA DE COURNOT

Trabalho apresentado ao Curso de Ciências Econômicas do Instituto de Ciências Sociais e Aplicadas (ICSA) da Universidade Federal de Ouro Preto como requisito para a obtenção do grau de Bacharel em Ciências Econômicas, sob orientação do Prof. Dr. Martin Harry Vargas Barrenechea.


Banca Examinadora:



Profa. Dra. Rosângela Aparecida Soares Fernandes



Prof. Dr. Victor Maia Senna Delgado



Prof. Dr. Martin Harry Vargas Barrenechea

Mariana, 17 de julho de 2018

Agradecimentos

O primeiro agradecimento é feito a Deus, que me deu forças em todo o momento e me ajudou a concretizar esse trabalho. Aos meus pais que me ajudaram e me acompanharam nesse processo sempre me estimulando a continuar nesse caminho; ao meu orientador Martin Harry que me ajudou desde o início dessa caminhada que com paciência me proporcionou adquirir amplo conhecimento; ao Gustavo Passos que com prontidão se dispôs a me ajudar com o código do NetLogo; aos professores Rosangela Aparecida Soares Fernandes e Victor Maia Senna Delgado que fazendo parte da minha banca se mostraram dispostos a contribuir com o meu trabalho; a Maria Cristina que com sua amizade me acompanhou nesse processo e a UFOP que com seu corpo de professores me proporcionou aprender cada vez mais.

*"E não vos conformeis com este mundo,
mas transformai-vos pela renovação
do vosso entendimento."
(Bíblia Sagrada, Romanos 12, 2)*

Resumo

O modelo oligopolista de Cournot gera controvérsias frente aos equilíbrios que podem ser encontrados em sua formulação para os modelos de aprendizagem social, em que as firmas aprendem com base em outras firmas, e de aprendizagem individual, em que as firmas aprendem com base em seu conjunto de estratégias, também chamada de "aprender fazendo". Em seu artigo [Vriend \(2000\)](#) argumenta que a aprendizagem social irá alcançar nível de equilíbrio Walrasiano enquanto a aprendizagem individual irá alcançar nível de equilíbrio de Nash. No trabalho de [Yildizoglu \(2002\)](#) é argumentado que os resultados encontrados pelo autor anterior são devidos aos parâmetros específicos utilizados para a função de lucro. Os mesmos autores fizeram uso de equações para encontrarem as soluções. Os resultados aqui encontrados com o uso da modelagem baseada em agentes, que aproximam os comportamentos dos agentes do comportamento real dos seres, e que apresentam certo nível de racionalidade por aprenderem por meio do algoritmo de aprendizagem conhecido como Algoritmo Genético, mostram que mesmo sob os parâmetros específicos de [Vriend \(2000\)](#), o equilíbrio encontrado no modelo de aprendizagem individual não converge totalmente para o equilíbrio de Cournot e o modelo social não converge para nenhum dos equilíbrios. Com a mudança dos parâmetros das funções, ambas as aprendizagens convergiram para o equilíbrio Walrasiano. O número de firmas presente no mercado também afeta o tipo de equilíbrio alcançado pelas mesmas.

Palavras-chaves: Cournot, Algoritmo Genético, Nash, Walrasiano.

Abstract

The Cournot oligopoly model creates controversies when it is related with the kind of equilibrium that can be found for the social (firms learn based in other firms) and individual learning (firms make "learning by doing"). In his article [Vriend \(2000\)](#) says that in social learning, the equilibrium will be Walrasian, while in individual learning the equilibrium will be Nash. In the work of [Yildizoglu \(2002\)](#), the author says that the results obtained by the first author can be explained by the specific parameters used in the functions. The results presented here make use of Agent Based Model which are close to humans behavior and the learning process is model by using Genetic Algorithm. The results show that using the specific parameters of [Vriend \(2000\)](#), the equilibrium in individual learning doesn't converge completely to the Cournot's equilibrium and in the social learning there's no convergence. With the change of parameters of functions, both learning models had convergence to the Walrasian equilibrium. The number of firms in the market have influence in the type of equilibrium.

Key-words: Cournot, Genetic Algorithm, Nash, Walrasian.

Lista de ilustrações

Figura 1 – Resultados encontrados por Vriend (2000)	38
Figura 2 – Aprendizagem Individual	39
Figura 3 – Aprendizagem Social	39
Figura 4 – Aprendizagem Individual e Social 1000 repetições	40
Figura 5 – Aprendizagem Social - 10, 20 e 60 firmas	42
Figura 6 – Aprendizagem Individual - 10, 20 e 60 firmas	42
Figura 7 – Aprendizagem Individual e Social - novos parâmetros das funções	43

Lista de tabelas

Tabela 1 – Parâmetros da modelagem	35
Tabela 2 – Parâmetros para o AG	35
Tabela 3 – Cordas binárias e seus <i>fitness</i> respectivos	36
Tabela 4 – Valores obtidos nas simulações	39

Sumário

Introdução	11
1 Equilíbrios em Concorrência Imperfeita	17
1.1 Oligopólios	17
1.2 Equilíbrio Cournot-Nash	17
1.3 Equilíbrio Walrasiano	20
2 Algoritmos Genéticos	23
2.1 Uma breve história da ciência da computação	23
2.2 O Algoritmo Genético	24
2.2.1 Operadores Genéticos	25
2.2.2 Cálculo dotness	28
3 Protocolo ODD	31
3.1 ODD - Aprendizagens Individual e Social	31
3.1.1 Visão Geral	31
3.1.1.1 Propósito	31
3.1.1.2 Entidades, variáveis de estado e escalas	31
3.1.1.3 Visão geral e cronograma de processos	31
3.1.2 Conceitos de Design	32
3.1.2.1 Princípios Básicos	32
3.1.3 Detalhes	34
3.1.3.1 Inicialização	34
3.1.3.2 Dados de Entrada	34
3.1.3.3 Submodelos	34
4 Análise de Resultados	38
4.1 Análise dos Modelos de aprendizagens social e individual	38
Conclusão	44
Referências	47
APÊNDICE A Aprendizagem Invididual	48
APÊNDICE B Aprendizagem Social	54

Introdução

Os oligopólios possibilitam a interação estratégica entre as firmas que o compõem e para isso foram desenvolvidos modelos dentro da teoria dos jogos para propor resultados e convergências, como o modelo de Bertrand, modelo de Stackelberg e o modelo de Cournot. O último modelo citado, foi uma das primeiras análises formais de comportamento estratégico em oligopólios, que teve sua formulação anterior ao equilíbrio de Nash e será tratado no presente trabalho. Para essa busca do equilíbrio de Nash do jogo de Cournot, as firmas determinam simultaneamente a quantidade que será produzida e posteriormente os preços e lucros serão determinados, no qual o preço de mercado estará acima do custo marginal e, conseqüentemente, as firmas obtêm lucros positivos.

Outro tipo de equilíbrio muito trabalhado dentro do contexto oligopolista é atribuído ao economista francês Leon Walras. O economista deu base para investigações modernas sobre questões bem amplas, que se trata de um modelo de toda a economia que consiga mostrar as possíveis conexões existentes entre vários mercados e agentes econômicos, assim como a interação de muitas firmas em um mesmo mercado. Para isso, o economista busca representar a economia com várias equações simultâneas que conduzem à compreensão de relações para a análise do equilíbrio geral, no qual é considerado que as firmas são tomadoras de preço e produzem no ponto em que o custo marginal se iguala ao preço.

Para encontrar equilíbrios (situações nas quais ofertantes e demandantes estão satisfeitos com os resultados obtidos no mercado), são propostos modelos que possuem uma população de agentes que interagem entre si em um espaço de tempo em que há a procura do equilíbrio de Cournot-Nash ou possíveis convergências para outros tipos de equilíbrio.

Os modelos citados acima estão relacionados com a teoria dos jogos tradicional, que trata os agentes como sendo racionais seguindo os axiomas de racionalidade encontrados na microeconomia. Em 1973, John Maynard Smith e G. R. Price escreveram primeiramente sobre a aplicação da teoria dos jogos para evolução. Nos modelos evolutivos não há o pressuposto de que os agentes se comportam de maneira racional, mas jogam do modo que seu gene o deixa programado. Quanto mais bem-sucedida é uma estratégia de um jogador na população, mais suscetível é a sua sobrevivência e há uma grande probabilidade que a estratégia seja passada para as gerações futuras.

Um dos conceitos desenvolvidos por [Smith e Price \(1973\)](#) é o de "Estratégia Evolutivamente Estável" (EEE), que é uma estratégia que se todos os membros da população a adotassem, nenhuma estratégia mutante conseguiria invadir a população.

Outra perspectiva feita na microeconomia é a de interação entre os agentes. Em um contexto de contato entre os agentes, por presenciarem diferentes estratégias de outros indivíduos,

ou até mesmo em sua própria população de estratégias, estão sujeitos a adaptação para buscarem melhores resultados em seus problemas. Assim, alguns ramos da teoria econômica estão fazendo uso de agentes artificiais adaptativos, já que as interações de mercados e dos agentes envolvidos podem modificar as crenças e ações dos mesmos.

Muitas das modelagens tradicionais são feitas por meio de cálculos diferenciais (a variável em questão é uma função), conseqüentemente os modelos tendem a ficar mais simples para que possam ser resolvidos matematicamente, sendo assim mais limitados. A presença da modelagem em computadores facilitou esse processo, pois as limitações colocadas pela abordagem matemática são removidas. Essa estrutura representa o sistema individual e os seus respectivos comportamentos. De acordo com [Railsback e Grimm \(2012\)](#), as MBAs (modelagens baseadas em agentes) são modelos em que os indivíduos e os agentes são colocados como únicos e como entidades autônomas que interagem entre si e em seu ambiente. Também proporciona o estudo de sistemas dinâmicos que surgem do comportamento dos indivíduos.

O comportamento dos agentes pode ser modelado e observado por meio do MBA, e assim sendo entendido. Proporciona a inclusão de processos que são importantes, mas que são de alta complexidade para serem incluídos em modelos simples, como os de equações diferenciais. Esse tipo de modelagem busca construir modelos em que os agentes, de certo modo, se comportam como pessoas e como se adaptam, mostrando como fornece uma abordagem diferenciada para estudos econômicos.

A MBA dá margem a um outro tipo de estudo, em que são abordadas as questões evolutivas e adaptativas dos agentes modelados. Alan Turing (1912-1954), John Von Neumann (1903-1957) e Norbert Wiener (1894-1964) foram os primeiros cientistas da computação e buscavam colocar nos programas de computador com inteligência habilidades semelhantes à vida de se replicar e com a capacidade de aprender e controlar o ambiente em que vivem.

Nos modelos de aprendizagem, os agentes são combinados de maneira aleatória com outros membros da população. Os jogadores se baseiam em suas experiências passadas e analisam como podem melhorar, e frente a isso pode-se notar um grau de racionalidade, mas não como na teoria dos jogos tradicional, já que os agentes não tendem a distorcer as suas estratégias para afetar os demais jogadores.

Concernente aos modelos de aprendizagem, John Holland, em 1960, foi responsável por inventar o Algoritmo Genético (AG). Em 1975 lançou o livro *Adaptation in Natural and Artificial Systems* (Adaptação em sistemas naturais e artificiais), em que faz a apresentação do AG como uma abstração da evolução biológica e fornece uma estrutura teórica para adaptação sob o uso do AG. Basicamente, o algoritmo formulado por Holland é um método para se mover de uma população de cromossomos para uma nova população usando operadores que foram inspirados na genética e que, de acordo com [Mitchell \(1998\)](#), são expressos por: seleção, crossover, mutação e inversão. No presente trabalho serão usados os operadores genéticos de

seleção, crossover e mutação que abrangem o uso do AG para que a racionalidade, ou seja, a capacidade de aprendizagem dos agentes seja representado de modo a transparecer a evolução dos mesmos no meio.

O AG estuda o tipo de comportamento adaptativo pelos quais os agentes irão passar. O algoritmo é usado como ferramenta de otimização, mas vale ressaltar que busca mostrar, no contexto econômico, os aspectos de uma população econômica que é demonstrado pelos agentes que possuem certo grau de racionalidade.

No contexto do desenvolvimento computacional para buscar similaridade humana, existe a também chamada Inteligência Artificial (IA). O desenvolvimento dessa área de pesquisa teve seu desdobramento no século XX. Nas indagações econômicas, o uso da IA enfrentou alguns empecilhos para seu desenvolvimento. Quando iniciaram o aumento dos usos dos IAs, estes eram voltados para as áreas que os haviam desenvolvido, como a engenharia, desenvolvido por [Benaroch \(1996\)](#), e os economistas hesitaram em colaborar com as pesquisas nessa área, mas aquelas que foram feitas tiveram reconhecimento pelo valor potencial de suas técnicas e que ganham cada vez mais força na economia.

Muitos trabalhos foram desenvolvidos com base na capacidade de aprendizagem dos agentes para obterem melhores resultados para suas firmas e que também fizeram uso do AG para a adaptação dos agentes. Exemplos são os trabalhos desenvolvidos por [Arifovic e Maschek \(2006\)](#), [Dawid e Kopel \(1998\)](#), [Bischi, Lamantia e Radi \(2015\)](#), [Vallée e Yigit \(2009\)](#) e [Vriend \(2000\)](#).

Conjuntamente, [Vriend \(2000\)](#) em seu artigo fez uso do AG em sua forma de aplicação para modelos econômicos no contexto evolucionário de aprendizagens social e individual. O modelo de aprendizagem social diz respeito a um conjunto de firmas em que cada uma possui uma regra saída, isto é, o nível de produção adotado por cada uma, e que ao fim de um período determinado aprendem com as estratégias adotadas pelas outras firmas, observam aquelas que encontraram melhores resultados e adaptam as estratégias utilizadas para essas soluções para si mesma, conhecido como "aprendendo com os outros".

Já o modelo de aprendizagem individual é relativo a um conjunto de firmas que possuem conjuntos de estratégias próprias e que no fim de certo tempo, após seus conjuntos de estratégias terem sido testados e os resultados de cada estratégia terem sido obtidos de acordo com o nível de saída determinado, as firmas irão escolher as melhores estratégias com as melhores soluções e irão aprender com base nelas. A firma aprende com base em suas próprias estratégias passadas, conhecido como "aprender por fazer".

Com essas especificações, [Vriend \(2000\)](#) relatou que no modelo de aprendizagem social, o nível de saída médio das firmas converge para o equilíbrio Walrasiano, enquanto no modelo de aprendizagem individual o nível de saída médio das firmas converge para o equilíbrio Cournot-Nash. O equilíbrio Walrasiano, conhecido também como modelo de equilíbrio

econômico geral, pode ser de nido como: dado um valor inicial de recursos, com as preferências que os consumidores possuem, pode-se chegar a preços e quantidades de equilíbrio nos mercados de bens e produtos.

O modelo de equilíbrio geral oferece uma clara conexão entre os mercados de todos os bens, assim como proporciona o exame de questões mais profundas sobre as relações dos mercados e as inúmeras rmas dentro de um mesmo mercado. As relações de troca são garantidas para os tipos de mercado. Uma consideração a ser feita é que segundo [Nicholson e Snyder \(2011\)](#) todo equilíbrio Walrasiano também é Pareto E ciente, ou seja, ninguém pode melhorar de situação sem que alguém piore de situação.

O equilíbrio de Nash diz respeito a uma situação, a qual pode ter dois ou mais jogadores (no caso de oligopólio os jogadores sendo representado pelas rmas), nenhum jogador tem estímulo para mudar de estratégia depois que o equilíbrio de Nash para a dada situação é de nido.

Por outro lado, [Arifovic e Maschek \(2006\)](#) questionaram os resultados obtidos por [Vriend \(2000\)](#), em que con rmaram os resultados obtidos para o modelo de aprendizagem social, mas questionaram a convergência obtida para o modelo de aprendizagem individual. As possíveis diferenças de convergências nos diferentes modelos podem estar relacionadas a memória que as rmas possuem de suas próprias estratégias, e no caso da aprendizagem social não há esse histórico. No trabalho feito por [Arifovic e Maschek \(2006\)](#), os autores variaram elementos do algoritmo que utilizaram e chegaram a conclusão de que a possível convergência para o equilíbrio de Cournot-Nash pode ocorrer por conta do modo específico como o desempenho das regras de produção são avaliados e, assim como colocado por [Vallée e Yld zgu \(2009\)](#), as funções específicas utilizadas no trabalho de [Vriend \(2000\)](#).

Na análise dos resultados de nossas simulações, concluímos que a convergência para o equilíbrio de Cournot-Nash se deve a duas coisas: a maneira específica pela qual o desempenho das regras de produção é avaliado juntamente com uma especificação de função de custo específica.

Tal como pode ser notado, houve uma contradição entre [Vriend \(2000\)](#) e [Arifovic e Maschek \(2006\)](#) sobre as convergências dos equilíbrios. Segundo [Vallée e Yld zgu \(2009\)](#), [Vriend \(2000\)](#) obteve seus resultados de convergência por ter usado parâmetros específicos para as funções de demanda inversa e de custo total. No artigo desenvolvido por [Vallée e Yld zgu \(2009\)](#), o autor corrobora a convergência para o equilíbrio Walrasiano no contexto da aprendizagem social que ocorre por conta do chamado efeito de despeito, no qual o agente escolhe uma ação que o prejudica, mas que prejudica mais os outros indivíduos, ou seja, o contrário de altruísmo. Tal efeito torna o equilíbrio Walrasiano estável.

Segundo [Vallée e Yld zgu \(2009\)](#) o efeito de despeito, que aparece na aprendizagem social, in uência na convergência dos resultados por fazer com que as empresas não prestem

atenção em seus próprios lucros e atrapalha o surgimento da dinâmica no modelo que busca melhores respostas e que assegura a estabilidade evolutiva do equilíbrio de Cournot, já que neste caso as firmas olham para o seu próprio lucro e buscam maximizá-lo.

O efeito de despeito não surge no modelo de aprendizagem individual, já que nesse modelo as firmas não observam as ações das demais que estão presentes no mercado e consequentemente seu processo de aprendizagem não se baseia em imitação das outras firmas e o efeito não ocorre, ou seja, as firmas se preocupam somente com o seu lucro não havendo desvio de atenção. Em seu artigo [Vriend \(2000\)](#) também leva em consideração a presença do efeito de despeito e atribui a diferença de convergências de equilíbrio a esse efeito.

De acordo com [Vallée e Yildiz \(2009\)](#) a convergência para o equilíbrio de Nash ocorre quando as firmas tem a oportunidade de encontrarem suas melhores estratégias antes que os operadores do AG comecem a ter efeito, ou seja, as firmas testam os resultados obtidos em seus conjuntos de estratégias uma por uma. Os mesmos autores também colocaram que o desenvolvimento da teoria dos jogos evolucionária contribuiu para o debate sobre os equilíbrios devido ao melhor ajustamento da dinâmica dos agentes.

No trabalho de [Arifovic e Maschek \(2006\)](#) é argumentado que no modelo de aprendizagem individual há o efeito de despeito, mas o mesmo não tem influência sobre a aprendizagem.

Ao buscar-se as quantidades ótimas para o modelo oligopolista de Cournot, é possível o encontro de diferentes resultados que convergem para diferentes níveis de equilíbrio sob a perspectiva de diferentes modelos de aprendizagem. Essa questão é muito abordada e gera controvérsias econômicas devido a diferentes soluções encontradas, como mostrado pelo contraste entre o trabalho de [Vriend \(2000\)](#) e [Arifovic e Maschek \(2006\)](#), e também [Vallée e Yildiz \(2009\)](#) coloca as diferenças devidas ao efeito de despeito.

O mercado oligopolista tem grande presença nas atuais estruturas de mercado, consequentemente atraindo um olhar mais atento para que seja analisado de maneira mais clara. Os possíveis comportamentos que as firmas podem ter nesse mercado podem ser analisados por meio dos equilíbrios (níveis de produção as firmas optam por atuar) e que são analisados no presente trabalho. Outro fator preponderante a ser analisado é o fato das firmas terem a capacidade de aprenderem com seus concorrentes ou até mesmo de aprenderem com base em suas experiências passadas levando a ideia de evolução no contexto biológico e que pode ser representado por um algoritmo que simula a aprendizagem e evolução dos indivíduos. Para que isso fosse feito e melhor estudado, foi utilizado o Algoritmo Genético.

Diante disso, o presente trabalho busca, por meio da modelagem baseada em agentes (MBA), com o uso do software NetLogo, replicar os resultados obtidos por [Vriend \(2000\)](#) com os mesmos parâmetros utilizados pelo autor nas funções de lucro, demanda inversa e custo total. O algoritmo de aprendizagem que será utilizado em ambos os modelos são os Algoritmos Genéticos.

O Algoritmo Genético proporciona aos agentes do modelo um certo grau de racionalidade, pois os mesmos aprendem com base em suas melhores estratégias passadas e nas estratégias dos demais agentes do modelo e escolhem aquelas com melhor resultado para reprodução em vista de que as próximas gerações forneçam soluções melhores. Com isso, as soluções do AG podem levar a resultados diferentes dos encontrados por [Vriend \(2000\)](#), como diferentes convergências para os equilíbrios dos modelos a serem analisados, devido a racionalidade.

O trabalho de [Vallée e Yildiz \(2009\)](#) colocou que os parâmetros específicos das funções utilizadas por [Vriend \(2000\)](#) e o efeito de despeito contribuíram para os resultados encontrados. Esse fato será abordado no presente estudo, de modo que, será adotada a mesma função de demanda inversa utilizada em [Vriend \(2000\)](#) em que os parâmetros da função serão posteriormente alterados. A demonstração de como os equilíbrios são encontrados, ou seja, suas respectivas derivações também são mostradas de maneira mais detalhada, o que não ocorreu nos trabalhos anteriores.

O uso da modelagem baseada em agentes traz uma abordagem diferente para o problema que havia sendo formulado e discutido em relação aos equilíbrios obtidos nos diferentes tipos de aprendizagem no modelo oligopolista de Cournot. A MBA, ao procurar imitar o comportamento humano, no modelo que será aqui desenvolvido irá buscar o real comportamento das firmas levando em conta sua racionalidade ao produzir com estratégias distribuídas aleatoriamente no início do processo de produção, mostra como o equilíbrio a ser encontrado estará mais próximo da real dinâmica nesse tipo de mercado.

O trabalho foi organizado em 4 capítulos. O primeiro irá introduzir o conceito de oligopólio e os equilíbrios de Nash e Walras com suas respectivas derivações para as funções tratadas no presente trabalho para o modelo de Cournot. O segundo capítulo trata sobre os algoritmos genéticos, discorrendo sobre como começaram os trabalhos na inteligência artificial e como o algoritmo funciona. O terceiro capítulo é referente aos protocolos de [OODR \(Overview, Design Concepts e Details\)](#) que especificam como os modelos de aprendizagem social e individual foram implantados no NetLogo. E o último capítulo faz a análise dos resultados obtidos nas simulações dos modelos de aprendizado.

1 Equilíbrios em Concorrência Imperfeita

1.1 Oligopólios

De acordo com [Nicholson e Snyder \(2011\)](#), o oligopólio é caracterizado como um mercado que possui poucas firmas, mas não só uma, como é característica do monopólio. Como possuem algumas firmas nesse mercado, existe a possibilidade de se analisar a interação estratégica entre elas com o uso da teoria dos jogos. Nesse tipo de mercado os preços são determinados de acordo com o modo que a firma compete no mercado, se é mais agressiva ou não.

A concorrência imperfeita, que é uma estrutura de mercado localizada entre a concorrência perfeita e o monopólio (ausência de concorrência), é uma das características do oligopólio. As firmas oferecem produtos similares ou iguais e os preços são menores do que no monopólio, mas são maiores do que em um estado de concorrência perfeita.

A base da teoria dos jogos possibilitou o desenvolvimento de modelos que ilustram o comportamento do mercado oligopolista, visto em [Osborne \(2004\)](#), como o modelo de Bertrand, Cournot e Stackelberg. No presente trabalho e como será demonstrado na próxima seção, é tratado do modelo de Cournot.

No modelo de Bertrand existem duas firmas que produzem produtos idênticos com um custo marginal constante. As firmas, no primeiro momento, escolhem os preços que irão oferecer no mercado de maneira simultânea e posteriormente decidem qual quantidade ótima será produzida, e conseqüentemente o lucro auferido no período.

No modelo de Stackelberg existem duas firmas que escolhem a quantidade que será produzida e produzem produtos idênticos. A característica que difere esse modelo da simultaneidade das escolhas de preços ou quantidades como é visto no modelo de Bertrand e no modelo de Cournot, é que existe a ideia de uma firma líder e de uma firma seguidora. Primeiramente a firma líder escolhe a quantidade ótima de saída do produto que será produzida e então posteriormente a firma seguidora irá determinar seu nível de saída com base na observação da quantidade determinada pela firma líder.

1.2 Equilíbrio Cournot-Nash

O jogo oligopolista de Cournot teve seu nome originado em Antonie Augustin Cournot (1801-1877) que foi um economista e matemático francês responsável por iniciar a sistematização da escrita econômica por meio de funções ao descrever suas áreas de estudo em demanda oferta e preço por meio de funções matemáticas. Cournot também buscou analisar os diferentes tipos de mercado, sendo um deles o mercado oligopolista, tendo formulado o modelo com observação em um duopólio.

O modelo desenvolvido por Cournot é um tipo de modelo dentro da economia utilizado de modo a demonstrar uma estrutura da indústria na qual as firmas competem com base na quantidade produzida.

Segundo Osborne (2004) os jogos estratégicos possuem certa estruturação, dada da seguinte maneira:

- Jogadores: composto por um conjunto de agentes;
- Ações: cada agente possui um conjunto de comportamentos;
- Preferências: os indivíduos possuem relações de preferências frente a suas ações.

De acordo com os itens que compõem a estruturação de um jogo estratégico, o modelo oligopolista de Cournot pode ser adequado como segue:

- Jogadores: As firmas;
- Ações: As firmas possuem conjuntos de ações que são representados pelo conjunto de saídas possíveis dados por números não negativos;
- Preferências: As representações das firmas são feitas pelos lucros encontrados por meio de:

$$p_i = P(Q)q_i - C_i(q_i) \quad (1.1)$$

para $Q = q_1 + q_2 + \dots + q_n = \sum_{i=1}^n q_i$ e q_i sendo as quantidades das firmas individuais.

A equação que representa $P(Q)$ é dada pela função de demanda inversa

$$P(Q) = a - bQ \quad (1.2)$$

No modelo oligopolista proposto por Cournot, existem n firmas que são compostas de um produto idêntico. Para a decisão de qual quantidade será produzida, as firmas decidem de maneira simultânea seus níveis de saída.

Para a demonstração de como o equilíbrio de Cournot-Nash é encontrado no presente modelo com as mesmas equações utilizadas por Vriend (2000), tem-se:

$$P(Q) = a - bQ \quad (1.3)$$

$$CT(q) = K + kq \quad (1.4)$$

as funções de demanda inversa e de custo total, em que os termos a , b e c são os parâmetros da função e K e k são os custos de produção fixos e custos de produção marginal, respectivamente. Para isso, as firmas buscam maximizar o seu lucro, que é dado por meio da maximização de:

$$\begin{aligned} p(q) &= Pq - CT \\ p(q) &= [a + bQ^c]q_i - [K + kq] \end{aligned}$$

em que $Q = \sum_{i=1}^n q_i$. Como no equilíbrio de Nash as firmas compreendem que podem influenciar o preço de mercado por meio do seu nível de saída, as firmas produzem no ponto em que o custo marginal é igual a receita marginal. Para que esse equilíbrio seja encontrado, os seguintes cálculos são feitos:

$$\begin{aligned} \frac{\partial p}{\partial q} &= [a + bQ^c] + [bcQ^{c-1}]q_i - k = 0 \\ bQ^c + bcQ^{c-1}q_i &= k - a \\ Q^c + cQ^{c-1}q_i &= \frac{k - a}{b} \end{aligned}$$

Como $q_i = Q/n$:

$$\begin{aligned} Q^c + cQ^{c-1} \frac{Q}{n} &= \frac{k - a}{b} \\ Q^c + c \frac{Q^c Q}{n} &= \frac{k - a}{b} \\ Q^c \left(1 + \frac{c}{n}\right) &= \frac{k - a}{b} \\ Q^c &= \frac{k - a}{b \left(1 + \frac{c}{n}\right)} \\ Q &= \frac{k - a}{b \left(1 + \frac{c}{n}\right)} \\ Q^N &= \frac{k - a}{b \left(1 + \frac{c}{n}\right)} \end{aligned}$$

As saídas individuais no equilíbrio de Nash são dadas por

$$q_i^N = \frac{Q^N}{n}$$

sendo o número de firmas do mercado.

A Tabela 1, na página 34, da seção de ODD mostra os valores dos parâmetros utilizados e que resultam nos seguintes valores para o modelo de aprendizagem individual:

$$\begin{aligned}
 Q^N &= \frac{k a^{1-c}}{b(1+c)} \\
 &= \frac{0 (1 \times 10^{97})}{1.5 \times 10^{95} (1 + \frac{39.99999997}{40})} \quad 1 = (39.99999997) \\
 &= 37,694.6
 \end{aligned}$$

e

$$\begin{aligned}
 q_i^N &= \frac{37,694.6}{40} \\
 &= 942.364
 \end{aligned}$$

para o total da quantidade e para o nível de produção individual.

Com os valores obtidos acima, o lucro obtido por cada firma é de:

$$\begin{aligned}
 p &= [1 \times 10^{97} + 1.5 \times 10^{95} \times 37,694.6^{39.99999997}] \times 942.364 \quad [4.097 \times 10^{94} + 0 \times 942.364] \\
 p &= 1.25645^{85}
 \end{aligned}$$

O trabalho de [Vriend \(2000\)](#) encontra a quantidade total $Q^N = 39,928.1$ e a saída individual dada por $Q^N/n = 998.2$.

1.3 Equilíbrio Walrasiano

Leon Walras (1834-1910) desenvolveu um método que busca abranger diferentes mercados e suas interações. Segundo [Nicholson e Snyder \(2011\)](#), "seu método de representar a economia por um largo número de equações simultâneas formam a base para entender a inter-relação implícita na análise do equilíbrio geral". Assim, o conceito do equilíbrio geral é de se ter um modelo que capte os efeitos da mudança de um mercado em um outro mercado, ou seja, olhar as repercussões em ambientes dado oscilações em outro.

Walras publica em 1874 o livro *Elements d'économie politique pure* (Elementos de economia política pura) o qual foi responsável por introduzir o modelo de equilíbrio geral na economia. No modelo de equilíbrio geral proposto a moeda é neutra, em que a dotação inicial dos recursos, tecnologia e preferências dos consumidores são dadas e por meio disso é possível encontrar preços e quantidades de equilíbrios dos mercados.

O modelo de equilíbrio geral usa das seguintes atribuições: começa ao definir o número de mercadorias a serem incluídas no modelo (como o consumo, bens intermediários, insumos produtivos) e determinar os preços de equilíbrio dos bens e verifica a mudança de preços devido a mudança de condições presentes no modelo. É suposto que as firmas buscam a maximização de seus lucros por meio de suas funções de produção e dos preços (tanto de saída quanto de entrada). A utilidade é maximizada pelos indivíduos. Em alguns modelos também é incorporada a atuação do governo, como impostos e empréstimos.

Diante disso, para o equilíbrio Walrasiano a ser utilizado no presente trabalho, a seguinte função é maximizada:

$$p(q) = Pq - CT \quad (1.5)$$

$$p(q) = [a + bQ^c]q_i - [K + kq] \quad (1.6)$$

em que $Q = \sum_{i=1}^n q_i$. Considerando que nesse modelo as firmas são tomadoras de preço, é feito com que o preço se iguale ao custo marginal. As funções de preço e lucro são dadas pelas equações 1.3 e 1.4. Como são tomadoras de preço (este já é dado), não é feita a derivada de primeira ordem em relação à variável de preço. Assim, para que esse equilíbrio seja encontrado, os seguintes cálculos são feitos:

$$\begin{aligned} \frac{\partial p}{\partial q} &= a + bQ^c - k = 0 \\ bQ^c &= k - a \\ Q^c &= \frac{k - a}{b} \\ Q^W &= \frac{k - a}{b} \end{aligned}$$

Tal resultado implica que o nível de saída individual pelas firmas no equilíbrio Walrasiano é

$$q_i^W = \frac{Q^W}{n}$$

As derivações feitas, tanto para o equilíbrio de Cournot quanto para o equilíbrio de Nash são correspondentes a um modelo de n firmas.

A Tabela 1, na página 34, da seção de ODD mostra os valores dos parâmetros utilizados e que resultam nos seguintes resultados para a modelagem social:

$$\begin{aligned}
 Q^W &= \frac{k a^{1-c}}{b} \\
 &= \frac{0 (1 \times 10^{97})^{1-(39.99999997)}}{1.5 \times 10^{95}} \\
 &= 63,7385655
 \end{aligned}$$

e

$$\begin{aligned}
 q_i^W &= \frac{Q^W}{n} \\
 &= \frac{63,7385655}{40} \\
 &= 15934641
 \end{aligned}$$

para o total da quantidade e para o nível de produção social.

Esses valores levam ao lucro de

$$\begin{aligned}
 p &= [1 \times 10^{97} + 1.5 \times 10^{95} 637385655^{39.99999997}] \times 15931641 [4.097 \times 10^{94} + 0 \times 15931641] \\
 p &= 4.097 \times 10^{94}
 \end{aligned}$$

O trabalho de [Vriend \(2000\)](#) encontrou as mesmas equações acima, sendo os valores considerados de $Q^W = 80,2421$ e $Q^W/n = 20061$.

2 Algoritmos Genéticos

2.1 Uma breve história da ciência da computação

Alan Turing (1912-1954), John von Neumann (1903-1957) e Norbert Wiener (1894-1964) são exemplos dos primeiros cientistas da computação e tinham a grande motivação de embutir nos programas de computador a capacidade semelhante à vida de replicar e de adaptar, por meio do aprendizado, assim como foram grandes formuladores da Inteligência Artificial (IA), em que seu formulador inicial, Alan Turing, havia denominado como "Maquinário Inteligente". Turing acreditava que era possível colocar as máquinas de computação para resolverem problemas por meio de buscas no espaço de possíveis soluções que são guiadas por regras práticas. Segundo Turing, de acordo com [Copeland \(2004\)](#), "a atividade intelectual consiste principalmente em vários tipos de busca".

Os anos 80 foram palco do ressurgimento da pesquisa na área acadêmica de computação biológica. O primeiro campo foi o de redes neurais que teve suas primeiras informações aparentes em 1943 desenvolvidos por Warren McCulloch do MIT e de Walter Pitts da Universidade de Illinois. Vale ressaltar também as contribuições feitas por Von Neumann nessa área. Em seu trabalho [Neumann \(1951\)](#) compara os órgãos e as partes de um computador, dentre essas partes os neurônios dos indivíduos e as partes de uma máquina.

As redes neurais são modelos computacionais inspirados no sistema nervoso central de um animal e que tem a capacidade de realizar o aprendizado de máquina e o reconhecimento de padrões sendo também associado a sistemas de neurônios interconectados que simulam as redes neurais de um ser.

O segundo campo é o de aprendizagem em máquinas que é um ramo da ciência da computação e que evoluiu da inteligência artificial, em que os computadores aprendem sem serem necessariamente programados. Como esse campo evoluiu da IA, teve a contribuição dos mesmos cientistas citados anteriormente para seu desenvolvimento. O terceiro campo é o de computação evolucionária, no qual os algoritmos genéticos são o exemplo mais relevante segundo [MITCHELL, 1998](#)). Esta é um tipo especial de computação que tira inspiração dos processos naturais de evolução, observado por [Eiben, Smith et al. \(2003\)](#), o que nos remete ao feito por Turing ao buscar aproximar a computação dos processos naturais dos seres.

A maioria dos estudos, mesmo feitos por Turing, remetem-se a teoria Darwiniana da evolução, que oferece uma explicação para a diversidade biológica e seus mecanismos, assim como Maynard Smith ao desenvolver a teoria dos jogos evolucionária. Por meio da teoria de Darwin, é notável a colocação sobre os instintos básicos dos indivíduos de reproduzir e selecionar, em que a seleção natural favorece aqueles que competem pelos recursos de modo mais

e ciente. Essa abordagem vai de encontro aos ideais propostos pelos cientistas apresentados e se encaixa na ideia proposta pelo AG. Para Darwin a ideia é conhecida como a sobrevivência do fittest é assim abordada pelos modelos computacionais para a seleção dos melhores resultados da população. Turing sugeriu busca genética ou evolucionária e outros cientistas, como Breinermann, fizeram tentativas de otimização por meio da evolução e recombinação. Muitos trabalhos paralelos foram feitos e um deles inclui o de Holland, como será visto.

2.2 O Algoritmo Genético

De acordo com [Goldberg \(2006\)](#) os "Algoritmos Genéticos são algoritmos de busca baseados na mecânica da seleção natural e da genética natural". Baseando-se na seleção do mais forte, as cordas binárias que estruturam esse algoritmo se recombina de modo a gerar novos indivíduos (cordas). As informações passadas dos agentes são levadas em conta para a aprendizagem.

John Henry Holland (1929-2015) foi um cientista e professor dos Estados Unidos, formado em física pelo MIT (Instituto de Tecnologia de Massachusetts), em matemática pela Universidade de Michigan e em 1959 recebeu um PHD em ciência da computação na mesma instituição. Holland foi responsável por inventar em 1960 os Algoritmos Genéticos (AG) e foi desenvolvido por ele, seus estudantes e colegas da universidade de Michigan entre os anos de 1960 e 1970.

Em 1975, Holland lançou o livro *Adaptation in Natural and Artificial Systems* (Adaptação em sistemas naturais e artificiais), que mostra o AG como uma abstração da evolução biológica e uma forma de adaptação por meio do mesmo algoritmo. O AG move de uma população de cromossomos para uma nova população por meio dos operadores, inspirados na genética, de seleção, crossover, mutação e inversão. O mecanismo de inversão não é muito utilizado na comunidade acadêmica por não ter sido amplamente aceito, mas os demais são utilizados.

O AG pode ser melhor entendido da seguinte forma: imagine as células do corpo humano. Estas são compostas pelos cromossomos que são longas sequências de DNA que possuem muitos genes e outros tipos de sequências. Os genes são segmentos das moléculas de DNA que contêm um código para a produção dos aminoácidos e codificam a proteína, ou seja, a caracterizam. Cada código, ou característica, é representado por um alelo.

No AG essa ideia é repassada de modo que os cromossomos são as soluções candidatas para o AG e podem ser codificadas de diferentes maneiras. A forma como a codificação do cromossomo ocorre para se formarem as soluções candidatas para as melhores respostas é um fator de extrema importância para que o AG obtenha sucesso na busca. Uma das codificações mais usuais é a de codificação binária, a qual será usada no presente trabalho.

A codificação binária é uma cadeia de bits, ou seja, as cordas que representam o cromossomo assumem valores de 0 e 1. Por meio dessa codificação houveram extensões que sugeriram

uma nova forma de codificação, exemplos são o código cinza, que também é um sistema de codificação binária desenvolvido por Frank Gray (1887-1969) e garante que qualquer mudança no decorrer da aplicação do AG resulte que apenas um bit da corda seja modificado, e o esquema de codificação de Hills. Os genes são as representações dos bits, podendo ser únicos ou blocos de bits. Como o gene é responsável por codificar a proteína, atribuindo-lhe as características através dos alelos, no AG o que atribui a característica (a quantidade a ser produzida) são a sequência de bits.

Outros meios de codificação dos cromossomos são o de muitos caracteres e codificações de valor real, em que são utilizados caracteres do alfabeto em larga quantidade ou o conjunto dos números reais, e o método de codificação de árvore. Este se mostra mais abrangente, pois permite que o espaço de busca do AG seja mais aberto, mas um problema que pode ser encontrado é de que as árvores podem crescer muito e não favorecerem a formação de possíveis soluções com os melhores.

Diante disso, [Mitchell \(1998\)](#) propõe que a melhor codificação a ser utilizada é aquela que seja mais natural para o problema e que o AG possa usá-la, ou seja, as codificações devem ser melhor adaptadas de modo que o AG possa encontrar as soluções ótimas no espaço de busca.

2.2.1 Operadores Genéticos

A codificação, como foi dito, é muito importante para o sucesso do AG. Após se ter definido qual tipo de codificação será utilizada o próximo passo, ou seja, a codificação binária (corda binária de determinado comprimento sendo composta por bits de valores 0 ou 1), é fazer o uso dos operadores genéticos: seleção, crossover e mutação. O operador de seleção escolhe os cromossomos na população que poderão se reproduzir. O operador de crossover, que no caso é a recombinação, troca subparte de dois cromossomos para recombinar biologicamente seus alelos e criarem um novo indivíduo. O operador de mutação muda aleatoriamente o valor de um alelo em alguma parte do cromossomo. Por fim, o operador de inversão rearranja a ordem em que os genes são ordenados.

O operador de seleção envolve o processo de como o indivíduo será escolhido na população para criar descendentes para a nova (próxima) geração. A ideia de selecionar os indivíduos mais aptos é a esperança de que os filhos destes sejam mais aptos e que ofereçam melhores resultados para o problema em questão. Para que a seleção seja feita, deve haver um equilíbrio entre a seleção dos indivíduos mais fortes, que reduz a diversidade da população e o progresso da mesma, e a seleção de indivíduos não tão aptos, que resultará em um processo de evolução lento. Assim, o operador escolhe cromossomos na população para que se reproduzam. Quanto maior a aptidão do cromossomo, mais vezes é provável que seja selecionado para a cópia. Segundo Holland, a seleção foca na busca de subconjuntos que possuem adequação acima da média obtida. Para que os agentes sejam selecionados, existem alguns processos de seleção tomando-se como base os demonstrados por [\(MITCHELL, 1998\)](#):

- **Roleta e Amostra Estocástica Universal** roleta cada indivíduo possui sua aptidão que é estabelecida como uma parte da aptidão total média da população de indivíduos o qual é chamado de "valor esperado"(número de vezes que o indivíduo será escolhido para reprodução). O método possui a seguinte implementação segundo (MITCHELL, 1998):
 1. Na população, a soma total do chamado valor esperado do indivíduo, seja este valor dado por G ;
 2. Repetir de acordo com o número de componentes da população;

Assim, um número inteiro entre 0 e G é escolhido. Posteriormente é feita a soma dos valores esperados dos indivíduos até que a soma seja maior ou igual ao número inteiro escolhido e o indivíduo que tiver seu valor esperado acima da soma é selecionado.

Já o processo de Amostragem Estocástica Universal diminui o intervalo dos números inteiros possíveis, dado o valor esperado do indivíduo e a roleta é girada uma vez, não é girada de acordo com o número de indivíduos na população.

- **Escala Sigma** nesse processo, os indivíduos que possuem alta aptidão são permitidos a terem muitos descendentes. Diferente do método anteriores, o valor esperado é dado por meio de uma função de sua adequação, da média da população e do desvio padrão da mesma.
- **Elitismo**: esse método força o algoritmo a reter alguns dos indivíduos que possuem as melhores soluções em cada geração, pois os mesmos podem se perder sem o processo de seleção e não obterem bons descendentes pelo processo de crossover
- **Seleção de Boltzmann** utilizado um parâmetro que pode variar e que controla o processo de seleção de acordo com uma agenda.
- **Seleção de Classificação** visa impedir que a convergência para os resultados ocorra de maneira muito rápida. A classificação inicial ocorre por meio da aptidão de cada indivíduo e o valor esperado depende disso. Esse processo evita que a maior parte dos descendentes esteja relacionada a um pequeno grupo de indivíduos que estavam altamente aptos para reprodução.
- **Seleção de torneios** indivíduos são escolhidos de maneira aleatória da população. Funciona da seguinte maneira: um número r é escolhido entre 0 e 1. É estabelecido um parâmetro k , caso $r < k$, o processo de seleção toma os indivíduos para formarem descendentes, caso contrário, o menos apto é selecionado. Os indivíduos selecionados são devolvidos a população e podem estar sujeitos a nova seleção.
- **Seleção de Estado Estacionário** somente alguns indivíduos são substituídos a cada geração, nesse caso alguns indivíduos menos adaptados são substituídos por novos agentes por meio do crossover

O critério avaliado pela seleção é o *fitness* de cada cromossomo. Quando o modelo é formulado, existe uma variável que serve como medida para as demais, sendo que quanto melhor seu resultado, maior é o *fitness*. Este, gera um sinal diferente de acordo com a guia do AG para a evolução das soluções do problema. Os critérios para essa medida mudam continuamente na medida que os indivíduos evoluem e conseqüentemente remete a ideia da evolução que busca um conjunto de possibilidades que está passando por processos constantes de mudança. A medida atribui uma pontuação a cada cromossomo na população atual.

Segundo a teoria de Darwin, as regras da evolução são as que as espécies evoluem por meio da variação randômica, em que posteriormente ocorre a seleção natural na qual os mais aptos tendem a sobreviver e a se reproduzir, enviando seu material genético para as próximas gerações.

Após a seleção dos indivíduos que serão escolhidos para reprodução, é aplicado o operador de *crossover*. O tipo mais comum utilizado é o de ponto único, que será utilizado no AG deste trabalho, em que uma localização (alelo) no gene é escolhida de maneira aleatória e as subsequências entre dois cromossomos são trocadas para que seja criado um descendente com material genético de ambos. Para *Holland*, o *crossover* faz a união daqueles que tem alta aptidão juntos na mesma cadeia para criar *fitness* mais altos.

Alguns dos usuários do AG preferem fazer o uso do *crossover* de dois pontos em que duas posições dos cromossomos são escolhidas de maneira aleatória e os segmentos são alterados. Esse tipo de *crossover* pode combinar mais esquemas do que o citado anteriormente.

Por meio de *crossover* descendentes são criados e com isso é iniciado o próximo passo: a mutação. Depois de criado um descendente com uma nova codificação, algum bit do gene é trocado, seja 1 por 0 ou 0 por 1. *Holland* coloca que a mutação garante que a diversidade genética gerada por esse operador nunca seja perdida em alguma localização do gene.

Com os operadores citados, o AG processa populações de cromossomos substituindo uma população pela outra de maneira sucessiva. Para a aplicação do AG na resolução de um problema os seguintes passos são seguidos:

- Definir a população de cromossomos;
- Calcular o *fitness* (aptidão) obtido de cada cromossomo;
- Aplicar os operadores de seleção, *crossover* e mutação, em que é estabelecida uma probabilidade para o *crossover* (crossover rate) ocorrer (nos casos em que não há cruzamento dos cromossomos, os descendentes são cópias de seus pais) e também é estabelecida uma probabilidade de mutação (mutation rate);
- A população é substituída pela nova;
- Os passos são repetidos.

A função dos AGs é buscar as melhores soluções, mas de acordo com Mitchell (1998) é importante fazer a diferenciação de que tipo de busca se trata. Existem os seguintes tipos de busca em ciências da computação:

1. Busca por dados armazenados: buscar as informações armazenadas em um computador, sendo um dos exemplos a pesquisa binária que busca com eficiência os registros;
2. Buscar caminhos para metas: encontrar soluções que se moverão de uma dotação inicial para um determinado objetivo;
3. Buscar soluções: encontrar de maneira eficiente uma solução para determinado problema em um amplo conjunto de soluções que se apresentam como candidatas.

O tipo de busca feito pelo AG é o dado pelo item 3. As soluções candidatas vão sendo criadas enquanto o algoritmo se desenvolve. É um método que encontra soluções ótimas por meio de uma parte dos candidatos a soluções.

O AG é visto como um método micro analítico da simulação de sistemas evolucionários. Os métodos padrões que buscam analisar a dinâmica evolucionária são feitos por métodos matemáticos que fazem uso de equações diferenciais e possuem suas limitações. Além de possuírem as limitações por conta da matemática, pois para que seja resolvido os sistemas de equações são formulados de maneira mais simples, esse tipo de análise só capta a dinâmica do sistema em nível global. Dizer isto é remeter a ideia de que detalhes que possam existir no modelo formulado não sejam captados, como um certo nível de racionalidade dos agentes e as decisões que tomam no ambiente. Além disso, essas modelagens tradicionais só captam a dinâmica global de um sistema em evolução. A simulação micro analítica feita pelo AG simula cada componente de um sistema em evolução e suas interações locais, sendo que as dinâmicas globais surgem dessas dinâmicas locais.

2.2.2 Cálculo dotness

Para que o número de possibilidades de cordas de bits seja encontrado (no caso o número de estratégias possíveis que os indivíduos podem assumir - número de cromossomos) o cálculo é feito por meio de:

$$2^l \tag{2.1}$$

em que l é o comprimento do gene (que aqui também será chamado de corda binária - assume valores de 0 ou 1).

O comprimento da corda binária é um fator importante para o significado do processo de aprendizagem das regras, pois quanto maior for o comprimento, o processo de procura das

mas é cada vez mais refinado. Para chegar mais perto da melhor estratégia no espaço de busca, as rmas usam de comprimentos maiores das cordas binárias. Para que uma rma aprenda, a mesma precisa fazer a aplicação de muitas regras antes de chegar a melhor delas, desse modo há um custo no processo de aprendizagem. Disso, pode ser percebido que os limites do espaço de busca da rma não dependem necessariamente do comprimento, mas sim de um processo de busca refinado.

O algoritmo não deve ser visto como um método de representação dos mecanismos de decisões exatas das rmas, mas sim como uma representação da aprendizagem e o processo de procura de estratégias por meio de experiências orientadas.

O AG, para [Holland \(1992\)](#), alcança resultados em um equilíbrio quase ótimo e baseia sua argumentação na analogia do problema do Bandido de Duas Armas (Two-Armed Bandit Problem). Usou esse modelo para mostrar como o AG aloca as amostras que possui aos intitulados esquemas (isto é, correspondem a conjuntos de genes que trabalham juntos para afetar alguma adaptação no organismo, descobrir evoluções e propagar as mesmas).

A ideia do problema é a seguinte: a um jogador são dadas N moedas para que jogue em uma máquina com os dois braços. Existem duas armas A_1 e A_2 , com taxas de payoffs $(1-4)^1$ e $(1-4)^2$ com variâncias A_1^1 e A_2^2 . Os processos de payoff são estacionários e os eventos das armas são independentes. O objetivo do jogador é maximizar o payoff na obtenção de informações das recompensas obtidas por cada arma. Assim, a medida que as informações são obtidas por meio da amostragem, a estratégia do jogador aumenta a probabilidade da amostra do braço melhor. Conseqüentemente uma estratégia quase ótima surge implicitamente, o que leva a maximização do desempenho e um melhor resultado alcançado.

Para a formulação do AG é necessário que uma função de fitness adequada seja definida. Caso isso não ocorra, a otimização da função pode levar a resultados não satisfatórios e dificulta a busca por um ótimo global.

Como foi dito, os cromossomos são compostos pelos genes e que possuem os alelos codificados em 1 ou 0. A questão é saber como calcular o fitness obtido de cada corda binária. Para o cálculo do valor de cada binária, será utilizado como referência o trabalho de [Yildizoglu \(2002\)](#) o qual faz tal demonstração de como chegar aos resultados:

- Sejam as estratégias codificadas como cromossomos G ,
- O comprimento da corda binária é dado por G ;
- Os valores decimais dos cromossomos correspondem a posição de sua estratégia em um espaço de busca compreendido entre $[0, 100\%]$.

- O espaço citado possui

$$D = \sum_{i=0}^{G-1} a_i k \cdot 2^i \quad (2.2)$$

estratégias estritamente positivas e igualmente espaçadas, em que k é o valor do bit da corda binária, zero ou 1.

- Cada cromossomo possui uma taxa correspondente no total no espaço dado por:

$$T = (C_i)_{10} \cdot \frac{1}{D} \quad (2.3)$$

em que o 10 representa os valores assumidos na corda binária, de 0 a 1. T é a taxa de cada cromossomo dentro do espaço de busca.

Com o cálculo de cada cromossomo, os operadores genéticos iniciam suas respectivas operações e o algoritmo genético é aplicado.

3 Protocolo ODD

3.1 ODD - Aprendizagens Individual e Social

3.1.1 Visão Geral

3.1.1.1 Propósito

O objetivo do presente modelo é investigar a convergência dos modelos de aprendizagem social e aprendizagem individual com o uso dos algoritmos genéticos que vão representar o comportamento adaptativo dos agentes na presença do chamado efeito de despeito (isto é, o jogador escolhe uma ação que prejudica a si mesmo, mas que prejudica mais ao outro jogador) e como tal efeito influencia nas convergências desse mercado, para o equilíbrio Walrasiano e o de Nash.

3.1.1.2 Entidades, variáveis de estado e escalas

O modelo possui como entidade (isto é, o que é representado no modelo) as *rmas* e as *mesmas* vivem em um mesmo mercado. A variável de estado é representada pelo custo de produção x_0 (K). O tempo é representado em dias, em que é representado um período de 100 dias e são feitas 50 simulações que representam as gerações. O algoritmo genético é aplicado ao m dos períodos múltiplos de 100 dias.

3.1.1.3 Visão geral e cronograma de processos

Nesse passo é tratada a dinâmica do modelo, que no caso são os processos que afetam as variáveis de estado das entidades, ou seja, o procedimento "go". As *rmas* terão seus períodos dados em 100 dias e serão formuladas 50 gerações resultando em um total de 5000 dias. Caso o tempo seja excedido o procedimento é interrompido. Será feito o cálculo das estratégias distribuídas aleatoriamente para as *rmas*.

No modelo de aprendizagem individual as 40 estratégias de cada *rma* serão calculadas e os lucros diários das *rmas* pela aplicação de `costdag` serão obtidos após a quantidade de saída de cada *rma* ser definida e o preço diário. Ao m do período de 100 dias, o algoritmo genético será aplicado e as 30 maiores estratégias que resultaram nas melhores soluções serão escolhidas para produção por meio do comando `codecover`. Após passarem pelo procedimento de `crossover` 10 novas estratégias terem sido criadas será aplicado o processo de mutação.

No modelo de aprendizagem social a estratégia de cada *rma* será calculada e será utilizada pela *rma* no decorrer dos 100 dias. Ao m do período, as 30 maiores estratégias, com base no *fitness*, dentre as *rmas*, serão escolhidas para reprodução e passará pelo processo de

observação e imitação pelas demais firmas. O algoritmo será aplicado e 10 novas estratégias são geradas. Depois de as novas estratégias terem sido criadas o processo de mutação será aplicado

3.1.2 Conceitos de Design

3.1.2.1 Princípios Básicos

A convergência da quantidade produzida aplicando-se o algoritmo genérico no modelo de aprendizagem individual para o equilíbrio de Nash ocorre por conta de que existem várias firmas nesse mercado e cada uma possui um conjunto de regras de saída, em que cada regra diferente é aplicada a um dia de produção, e percebem que conseguem influenciar o preço de mercado através de suas próprias saídas e acreditando que suas decisões de saída não influenciam a decisão de saídas das demais firmas que podem existir no mercado. Diante disso, a firma irá produzir no ponto em que o custo marginal é igual a receita marginal ($RMG = CMG$).

A convergência da quantidade produzida aplicando-se o algoritmo genético no modelo de aprendizagem social para o equilíbrio Walrasiano ocorre por que existem várias firmas nesse mercado (40) e que cada firma individual possui uma regra de saída diferente das demais, em que cada regra de saída é usada pela firma durante o período estabelecido, e tendem a se comportar como tomadoras de preço e produzem para o ponto em que seu custo marginal é igual ao preço de mercado ($CMG = P$).

A firma determina a quantidade a ser produzida através da regra de saída estabelecida pela corda binária. São considerados 100 períodos (dias) em que o algoritmo genético é usado ao fim deste e aos períodos múltiplos de 100 para modificar o conjunto de regras que uma firma individual tem em mente e no modelo de aprendizagem social mudar a regra que cada firma tem.

Quando o algoritmo genético é aplicado, para o modelo individual a firma olha o quanto bem fez no passado quando usou suas próprias regras e tomando como base o lucro obtido a cada dia em que foi usada uma corda binária diferente, sem olhar para as estratégias adotadas pelas demais firmas no mercado (essa atitude é adotada por todas as firmas nesse mercado, elas se baseiam em suas próprias estratégias). Por outro lado, no modelo social, quando o algoritmo é aplicado espera-se que as firmas olhem ao seu redor, e tendam a imitar, e recombina ideias de outras firmas que aparentam ser de sucesso. Quanto mais bem-sucedidas essas regras foram, maior a chance de que sejam selecionadas para o processo de imitação e recombinação, em que, no presente caso, o sucesso é determinado pelo lucro gerado por cada regra de saída.

Os presentes modelos se baseiam nas experiências feitas por [Vriend \(2000\)](#) e tem relação com a busca pelas soluções encontradas pelo autor, mas aqui é feita uma modelagem baseada em agentes.

Os resultados importantes a serem obtidos desse modelo estão relacionados a convergência da quantidade média obtida por cada firma ao equilíbrio de Cournot-Nash (usando os

métodos de cálculo desse tipo de oligopólio) no modelo individual e para o equilíbrio Walrasiano no modelo social. Esses resultados serão obtidos ao ser aplicado o comportamento adaptativo das firmas por meio do algoritmo genético o qual, ao ser implementado, irá usar os valores fornecidos por [Vriend \(2000\)](#) e que são mostrados na seção de Dados de Entrada.

Como foi dito, os agentes usam do comportamento adaptativo baseado nos algoritmos genéticos (o qual é descrito no capítulo de Algoritmos Genéticos). Os agentes terão mudanças em suas estratégias de produção. Do modo que é aplicado o AG, a alteração ocorre pelo método aplicado por este.

Ao fim do período de 100 dias, o algoritmo irá selecionar os melhores $parents$ resultantes das quantidades produzidas, reproduzi-los e mutar a corda binária em busca de melhores resultados. Nesse caso, os agentes tomam a decisão de qual quantidade será produzida e com base nessa decisão irão obter o preço de mercado e o lucro do dia. Os traços adaptativos dos agentes fazem com que os agentes busquem e utilizem as estratégias que retornam os maiores resultados, já que cada firma busca maximizar seu lucro.

Os agentes buscam maximizar seu lucro e este será avaliado no modelo como uma medida de *fitness*. Para ser obtido é feita a maximização da função de lucro da firma que é dependente da decisão de quantidade a ser produzida e o preço é obtido através da função inversa da demanda, logo também é dependente da decisão de produção. Como as firmas estão buscando tal maximização e as quantidades são decididas com a regra de saída de acordo com a corda binária que sofre mutação por meio do AG, essa medida de lucro se relaciona claramente com o comportamento adaptativo dos agentes.

O *fitness* irá mudar conforme a decisão de regra de saída se alterar. Assim, quando ocorre mutação é esperado que os valores obtidos também mudem. Os agentes não irão mudar seu modo de adaptação, pois o mesmo algoritmo será aplicado a cada período.

A previsão do modelo é feita com base em mecanismos de memória. A cada 100 dias, em que é aplicado o AG, a firma olhará para as estratégias adotadas a cada dia e aquelas decisões de saída que proporcionaram o maior *fitness* será escolhida para reprodução, que no caso aqui em questão serão feitas cópias idênticas dos pais, e sofrerá mutação para que resultados cada vez maiores sejam encontrados.

No modelo individual, as firmas acreditam que suas decisões de saída não afetam as decisões tomadas pelas demais firmas, já que cada uma olha somente para o conjunto de regras que possui e não busca opções de estratégia em outros, e também acreditam que as decisões que tomam influenciam o preço de mercado, logo se baseiam nisso para se estabelecerem. Já no modelo de aprendizagem social, as firmas produzem de modo a atingir o ponto em que o custo marginal é igual ao preço e se adaptam com base nos resultados obtidos por elas e pelas demais firmas. As mesmas irão se estabelecer com base no processo de cópia e recombinação das estratégias das demais companhias.

O modelo de aprendizagem individual, assim como no modelo de aprendizagem social, as *rmas* compõem o mesmo mercado, mas para os resultados a serem obtidos as *rmas* não possuem interação uma com a outra.

Os agentes passam pelo comportamento adaptativo por meio do AG. Este por sua vez possui uma combinação particular de elementos que é feita por meio da busca paralela baseada no conjunto de regras de saída com seleção estocástica dos melhores obtidos (melhor resposta de quantidade produzida), crossover estocástico e da mutação de acordo com (MIT-CHELL, 1998). Isso ocorre por conta da de nição de processos estocásticos, os quais são uma família dentre da teoria das probabilidades que representam variáveis aleatórias que evoluem em um sistema de valores com o tempo.

Para que a dinâmica interna do modelo seja observada, o nível de produção da *rma* deve convergir para o resultado obtido no equilíbrio de Nash Cournot, sendo o nível de saída agregado dado por $Q^N = 39,9281$ e com simetria para uma saída individual $Q_i^N = 9982$, para a aprendizagem individual. No modelo social espera-se que o resultado a ser obtido no equilíbrio Walrasiano seja $Q^W = 80,2421$ e com simetria para uma saída individual $Q_i^W = 20061$. As ferramentas necessárias para que tais resultados sejam obtidos originam da aplicabilidade dos algoritmos genéticos.

3.1.3 Detalhes

3.1.3.1 Inicialização

A inicialização do modelo será feita usando os valores de entrada obtidos de [Vriend \(2000\)](#) que serão colocados na seção Dados de Entrada. Nos dois métodos de aprendizagem são criadas 40 *rmas*, em que no modelo social cada *rma* possui uma única regra de saída, ou seja, uma corda binária, e no modelo individual cada *rma* possui um conjunto de 40 regras de saída. Cada *rma* é diferenciada pela quantidade a ser produzida a cada período e consequentemente se distinguem por meio das estratégias, corda binária, adotadas as quais são distribuídas aleatoriamente.

3.1.3.2 Dados de Entrada

Os dados de entrada utilizados nos dois modelos são dados pela Tabela 1.

3.1.3.3 Submodelos

No modelo aqui representado, há somente um submodelo: os algoritmos genéticos.

Algoritmo Genético - GA

Para a formulação do algoritmo genético serão usados os parâmetros apresentados na Tabela 2.

Tabela 1 – Parâmetros da modelagem

Termo	Variável	Valor
Função de demanda inversa	$P(Q)$	$a + bQ^c$
Parâmetro da demanda	a	1×10^{97}
Parâmetro da demanda	b	5×10^{95}
Parâmetro da demanda	c	39.99999997
Custos de produção	K	4.097×10^{94}
Custos de produção marginal	k	0
Número de rmas	n	40

Fonte: Elaboração Própria, com base em [Vriend \(2000\)](#)

Tabela 2 – Parâmetros para o AG

Termo	Valor
Nível de saída mínimo individual	1
Nível de saída máximo individual	2048
Codi cação das cordas de bits	Binária padrão
Comprimento da corda de bits	11
Número de regras do AG individuais	40
Número de regras do AG social	1
Taxa do AG	100
Número de novas regras criadas	10
Seleção	Torneio
Probabilidade de seleção	$\frac{1}{\text{tness}}$
Crossover	Ponto
Probabilidade de crossover	0.95
Probabilidade de mutação	0.001

Fonte: Elaboração Própria, com base em [Vriend \(2000\)](#)

O termo "torneio" se refere ao torneio de seleção para os melhores.

As rmas irão usar as expectativas baseadas na última quantidade do período para fazer o cálculo do tness de cada estratégia da população. Para avaliar cada regra a rma usa a dinâmica industrial, pois o valor da regra depende disso para um número de períodos e a taxa média de lucro do intervalo de tempo para a adequação da estratégia. Quando todas as estratégias são avaliadas, no caso o lucro obtido de cada uma, uma nova população é gerada por meio dos três processos: seleção, crossover e mutação.

Os valores do tness (lucro) e o espaço de busca serão feitos com base nas equações 2.2 e 2.3 da seção de algoritmos genéticos.

Consequentemente, temos para o presente modelo $2^{11} = 2048$ cromossomos, baseado em [Mitchell \(1998\)](#), de $G = 11$. Se $G = 11$, existem:

$$(1111111111)_{10} = 1:2^{10} + 1:2^9 + 1:2^8 + 1:2^7 + 1:2^6 + 1:2^5 + 1:2^4 + 1:2^3 + 1:2^2 + 1:2^1 + 1:2^0 = 2047 = D$$

Estratégias estritamente positivas igualmente espaçadas entre 0 e 100%

De acordo com tal formulação, os fitness obtidos de cada estratégia serão formulados:

$$C_1 = (0000000000)_2 = 0:2^9 + 0:2^8 + 0:2^7 + 0:2^6 + 0:2^5 + 0:2^4 + 0:2^3 + 0:2^2 + 0:2^1 + 0:2^0 = 0$$

$$C_2 = (0000000001)_2 = 0:2^9 + 0:2^8 + 0:2^7 + 0:2^6 + 0:2^5 + 0:2^4 + 0:2^3 + 0:2^2 + 0:2^1 + 1:2^0 = 1$$

$$C_3 = (0000000011)_2 = 0:2^9 + 0:2^8 + 0:2^7 + 0:2^6 + 0:2^5 + 0:2^4 + 0:2^3 + 0:2^2 + 1:2^1 + 1:2^0 = 3$$

$$C_4 = (0000000111)_2 = 0:2^9 + 0:2^8 + 0:2^7 + 0:2^6 + 0:2^5 + 0:2^4 + 0:2^3 + 1:2^2 + 1:2^1 + 1:2^0 = 6$$

e assim sucessivamente para a população de estratégias de cada rma e para cada estratégia individual para o modelo social.

Será desenvolvido da seguinte forma:

- reescalar linearmente o fitness para [0,1]
- colocar um valor respectivo para corda binária de comprimento igual a 11, sendo que o nível máximo de saída individual por rma é de 2048 quantidades e o mínimo é dado por 1. Assim é dado um exemplo de representação da corda binária e o valor atribuído a cada uma, como pode ser visto na Tabela .

Tabela 3 – Cordas binárias e seus fitness respectivos

Corda Binária	Fitness
00000000000	0
00000000001	1
00000000011	3
:	:
00000000111	6

Fonte: Elaboração Própria

- A rma possui conjuntos de cordas binárias a ser estabelecido por dia e que de nirá a regra de saída para o mesmo. No modelo social a rma usa de uma estratégia no decorrer do período.
- Quando todas as estratégias da população são avaliadas no modelo individual, os resultados do período são avaliados no modelo social, é feita a seleção e uma nova população é gerada por meio do processo de crossover e posteriormente a mutação dos bits das cordas binárias.

Seleção De todas as regras possíveis (2048) associadas ao número de cromossomos possíveis serão escolhidas as 30 regras, assim como feito no trabalho de [Vriend \(2000\)](#), mais adequadas dentre elas, as que resultam no maior lucro.

Crossover. com probabilidade $p_c = 0:95$ (crossover rate), formar dois descendentes que sejam cópias exatas de seus respectivos pais, pois nesse caso as rmas baseiam suas novas estratégias em experiências passadas e copiam as melhores regras de saída do próprio conjunto de regras da empresa.

Mutação: mutar os descendentes em cada locus (localização) com probabilidade $p_m = 0:001$ (mutation rate), mudando bit de 1 para 0 e vice-versa.

Depois desse processo são criadas 10 novos descendentes com o processo de crossover. O processo é repetido e o AG é aplicado a períodos múltiplos de 100. Serão criadas 50 gerações.

Na aprendizagem individual as rmas não observam as ações e os lucros de outras rmas, a dinâmica de aprendizagem não é baseada na imitação. Conseqüentemente, o efeito de despeito não desempenha um papel de maneira direta segundo [\(VALLÉE; YIGDIZO 2009\)](#). Já no modelo de aprendizagem social ocorre o contrário, as rmas observam os lucros das outras e sua aprendizagem é baseada na imitação e, conseqüentemente, há o efeito de despeito

4 Análise de Resultados

Os modelos de aprendizagem social e aprendizagem individual foram analisados de acordo com a modelagem baseada em agentes feita no programa NetLogo, tendo como base de utilização o livro de [Railsback e Grimm \(2012\)](#) para a estruturação do modelo.

A formulação do modelo no programa seguiu primeiramente os mesmos parâmetros para as funções de lucro utilizados por [Vriend \(2000\)](#) para que os resultados encontrados pelo autor na presença da modelagem baseada em agentes e com o uso do algoritmo de aprendizagem sejam aproximados. As convergências para os equilíbrios obtidos por [Vriend \(2000\)](#) foram dadas pelo gráfico abaixo:

Figura 1 – Resultados encontrados por [Vriend \(2000\)](#)

Fonte: [Vriend \(2000\)](#), p.[7]

A intenção é que os resultados obtidos por [Vriend \(2000\)](#) sejam comprovados ou não pelos agentes que buscam aproximar seu comportamento do humano tanto para o modelo de aprendizagem social quanto para o modelo de aprendizagem individual.

As análises dos diferentes tipos de aprendizagem serão feitas a seguir.

4.1 Análise dos Modelos de aprendizagens social e individual

Os modelos de aprendizagem individual e social serão analisados conjuntamente para que as comparações entre os modelos pudessem ser feitas. As primeiras modelagens foram feitas com os mesmos parâmetros das funções de lucro e demanda inversa usados por [Vriend](#)

(2000) (as funções utilizadas foram mostradas na seção Equilíbrio Cournot-Nash, seção 1.2 página 16).

No primeiro momento foram feitas duas simulações para cada um dos modelos de aprendizagem que podem ser vistos abaixo, respectivamente:

Figura 2 – Aprendizagem Individual

Fonte: Elaboração Própria

Figura 3 – Aprendizagem Social

Fonte: Elaboração Própria

Os dados de máximos e mínimos obtidos nas simulações são apresentadas na Tabela 4, abaixo (a média é calculada a partir dos valores de máximo e mínimo):

Tabela 4 – Valores obtidos nas simulações

Aprendizagem	Simulação	Máximo	Mínimo	Média
Individual	1	1809.475	876.775	1343.125
	2	1752.2	792.9	1272.55
Social	1	2039	167.2	1103.1
	2	2017	59.2	1038.1

Fonte: Elaboração Própria

A análise do gráfico de aprendizagem individual se mostra mais conclusivo para análise de convergência de equilíbrio do que o gráfico referente ao de aprendizagem social. A Figura 2

mostra menos variação dos valores obtidos pelas 40 rmas, em que estas obtiveram valores das quantidades médias mais próximos variando entre 790 e 1810. Já a Figura 3 mostra uma grande variação e diferença entre os valores obtidos por cada uma das rmas, tendo sua variação da quantidade média presente entre 58 e 2040.

A diferença em relação ao comportamento dos agentes nos diferentes modelos pode ser associada a possibilidade de memória que os agentes possuem. No modelo de aprendizagem individual, como as rmas aprendem com base em seu próprio grupo de estratégias, as mesmas observam as atitudes que tomaram no passado e replicam aquelas que forneceram os melhores resultados depois de terem testado todas as estratégias de seu conjunto. Para o modelo de aprendizagem social isso não ocorre, pois as rmas aprendem com base nas melhores estratégias das demais rmas, sem possuírem um "banco de dados" ao qual pode ter acesso e replicar as melhores estratégias.

A dispersão dos valores obtidos por cada uma das rmas no modelo de aprendizagem social também está relacionado "spite effect"(efeito de despeito). As rmas ao perceberem que ao obterem melhores resultados nesse mercado, suas estratégias estão sendo imitadas pela demais rmas que não obtiveram resultados tão bons e também sendo re combinadas com outras estratégias que também são boas, acabam prejudicando a si mesmas em vista de prejudicar as demais companhias para que possa aumentar seu lucro adotando estratégias piores do que aquelas que poderiam adotar.

Para um melhor aproveitamento dos dados e uma melhor análise foram feitas uma simulação para cada modelo em que houveram 1000 repetições. Os gráficos abaixo mostram a média de todas as repetições para que algum equilíbrio seja verificado:

Figura 4 – Aprendizagem Individual e Social 1000 repetições

Fonte: Elaboração Própria

O primeiro gráfico, corresponde a aprendizagem individual e o segundo corresponde a aprendizagem social. Como a dispersão do modelo social é maior que o do modelo individual, o mesmo apresenta um comportamento diferente do primeiro.

A média do modelo de aprendizagem individual como já era de se esperar é menos

dispersa. No primeiro momento há um crescimento em direção ao valor de 1000 (proximidade com o valor médio obtido nas simulações com valores dados pela Tabela 4) e com o avançar do período com mais repetições convergiu para os valores máximos que a simulação poderia obter, mas claramente excedendo os resultados que haviam sido encontrados previamente. Ou seja, de acordo com a média das 1000 repetições, as *rmas* do modelo individual alcançam certo equilíbrio entre a primeira e a última geração (dado entre os tempos de 20 e 100) em direção ao equilíbrio Walrasiano considerado por (VRIEND, 2000).

A média do modelo de aprendizagem social, assim como nas simulações para a mesma aprendizagem mostradas acima, não se mostra constante ou perto de algum equilíbrio, seja Walrasiano ou de Nash, mostrando que o modo como as *rmas* aprendem e o efeito de despeito prejudicam que todas as *rmas* alcancem algum equilíbrio que forneça bons resultados para todas. Mas nesse modelo, algumas *rmas* alcançam bons resultados de média de quantidade produzido auferindo bons lucros, enquanto outras permanecem em baixos níveis de produção não obtendo valores consideráveis de lucro.

Os resultados obtidos também podem ser influenciados pelo número de *rmas* que existem nesse mercado, sendo isso mais aplicável para o modelo de social. Caso existam muitas *rmas* nesse mercado, a possibilidade de convergência para um equilíbrio é mais complexa, já que as *rmas* estarão colocadas de maneira mais dispersa e a chance de cópia das melhores estratégias não será possível por conta da assimetria de informação que haverá no mercado, mas também como há mais *rmas* nesse mercado, as mesmas percebem que não possuem influência no preço de mercado e tendem a convergência para o equilíbrio Walrasiano, pois assumem que são tomadoras de preço. Em vista disso serão feitas novas simulações em que serão variadas a quantidade de *rmas* desse mercado.

Diante de confirmar ou não, o fato do número de *rmas* poder influenciar nos equilíbrios, foram feitas simulações com 10, 20 e 60 *rmas* para os modelos social e individual com os mesmos dados utilizados nas simulações anteriores, vistos nas figuras 5 e 6. O número de *rmas* assumido foi disposto de maneira aleatória de modo a ficarem mais em torno do número original de *rmas* anteriormente adotado de 40.

Na aprendizagem social em que é composto por 10, 20 e 40 *rmas* não houve convergência para nenhum dos equilíbrios especificamente, mas quando foi aumentado o número de *rmas* para 60 houve a clara convergência da quantidade produzida pelas *rmas* para o equilíbrio Walrasiano assim como para o modelo de aprendizagem individual para o mesmo número de companhias nesse mercado. Para a modelagem individual, quando o número de *rmas* foi codificado em 10 e 20, as *rmas* tenderam a convergir para o equilíbrio de Nash.

Diante disso, pode-se verificar que o número de *rmas* presente no mercado tem influência sobre com qual quantidade de equilíbrio as *rmas* irão produzir.

O resultado encontrado mostra que sob os parâmetros específicos das funções utiliza-

Figura 5 – Aprendizagem Social - 10, 20 e 60 rmas

Fonte: Elaboração Própria

Figura 6 – Aprendizagem Individual - 10, 20 e 60 rmas

Fonte: Elaboração Própria

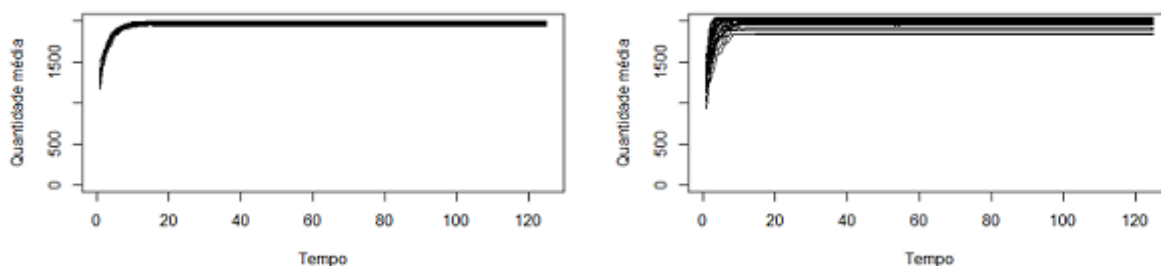
dos por [Vriend \(2000\)](#) as rmas no modelo individual para 10 e 20 rmas convergem para o equilíbrio de Nash, mas quando a quantidade de rmas nesse mercado aumentou para 40 e

60 o equilíbrio convergiu para o equilíbrio Walrasiano. Esse fato pode ser explicado pelo fato de que as firmas aumentam seu nível de competitividade diante de mais firmas no mercado e compreendem que o nível de saída das mesmas não pode afetar o preço de mercado.

No modelo social, com o número de firmas mais alto o efeito de despeito não teve tanto impacto quanto a convergência para um equilíbrio, pois como existem mais firmas a chance das empresas imitarem umas as outras se reduz por conta da distância que pode existir entre elas.

Uma nova modelagem para os dois tipos de aprendizagem com 40 firmas, mas com a mudança de parâmetro de $a = 10$, $b = 50$ e $c = 20$ podem ser vistas abaixo e que foram escolhidos de maneira aleatória com o objetivo de variar os termos sem especificidade:

Figura 7 – Aprendizagem Individual e Social - novos parâmetros das funções



Fonte: Elaboração Própria

Com o uso dos novos valores para as funções ambos os modelos convergiram para o equilíbrio Walrasiano, sendo que a aprendizagem individual convergiu com mais eficácia para o equilíbrio enquanto a aprendizagem social convergiu, mas não com tamanha precisão mostrando que o modo como as firmas aprendem influenciam na total convergência para um equilíbrio.

Conclusão

O presente trabalho buscou analisar os problemas de convergência gerados no modelo oligopolista de Cournot em relação a que tipo de equilíbrio os modelos de aprendizagem social e aprendizagem individual chegavam, sob primeiramente os parâmetros das funções de demanda inversa e lucro adotados mostrados na seção 1.2, na página 16, e fazendo modificações nos parâmetros utilizados para verificar se a condição colocada para os equilíbrios encontrados eram em virtude dos parâmetros específicos e como os resultados poderiam variar ao terem seus valores modificados.

O desenvolvimento do corpo do trabalho foi dividido de modo que a explicação de como os resultados foram encontrados fosse entendido. Assim, o primeiro capítulo mostra como os equilíbrios aqui estudados são encontrados para n firmas. Os valores usados para o equilíbrio Walrasiano e de Nash foram feitos com base nos parâmetros das funções dados pela Tabela 1 que se encontra na página 34 e substituídos na quantidade de equilíbrio encontrada.

Trabalhos anteriores mostraram que o modelo de aprendizagem individual convergia para o equilíbrio de Cournot-Nash, enquanto no modelo de aprendizagem social convergia para o equilíbrio Walrasiano. Alguns trabalhos confirmaram a convergência do equilíbrio de aprendizagem social, mas que não alcançaram os mesmos resultados para o equilíbrio individual.

O modelo de Cournot-Nash é visto como um modelo de visão parcial do mercado em que não capta todos os efeitos que podem ser vistos em um mercado. Já o modelo de equilíbrio Walrasiano é chamado também de equilíbrio geral, no qual busca verificar a interação das firmas em diferentes mercados, sendo assim um modelo de visão geral em que o equilíbrio das firmas é afetado por todas as demais que compõem o mercado.

Para a análise dos possíveis equilíbrios a serem alcançados no modelo oligopolista de Cournot, o presente trabalho usou da modelagem baseada em agentes, a qual é uma ramificação da inteligência artificial que busca replicar o comportamento humano, sendo aqui utilizado para replicar o comportamento das firmas presente no mercado oligopolista. O algoritmo de aprendizagem utilizado foi o de algoritmos genéticos, o qual busca na genética seu operadores de seleção, *CROSSOVER* e mutação para tornar o modelo evolucionário em que as firmas aprendem com os melhores resultados obtidos por suas gerações anteriores.

Os trabalhos formulados por todos os autores anteriormente citados geraram controvérsias em relação aos resultados encontrados, mas faziam o uso de equações e não captavam todo o comportamento dos indivíduos no mercado como é feito pela modelagem e com o uso do algoritmo de aprendizagem que proporciona aos agentes do modelo certo nível de racionalidade.

O algoritmo utilizado possibilitou que as firmas fossem testadas em períodos de 100

dias e ao fim deste os operadores trabalharam de modo a evoluir a população das firmas em busca de melhores resultados. A todos os períodos múltiplos de 100 o algoritmo foi aplicado a fim de gerar 50 gerações.

O uso da modelagem baseada em agentes junto aos algoritmos genéticos fornece uma perspectiva diferente para a análise do equilíbrio pelas firmas. Como os agentes buscam demonstrar a real atitude das firmas no mercado e como as mesmas podem aprender, tanto por meio de suas próprias estratégias (aprender por fazer), quanto aprender com as melhores estratégias de outras firmas (aprender de outros), os resultados aqui encontrados tendem aos reais equilíbrios que podem ser obtidos em mercados reais.

A modelagem feita no trabalho usou do *software* NetLogo e os códigos feitos para a formulação dos modelos podem ser visto nos apêndices A e B. A descrição de como o modelo foi pensado é descrito pelo ODD (*Overview, Design Concepts and Details*).

Os resultados obtidos geram controvérsias aos resultados que eram esperados durante a formulação do mesmo. Primeiramente era esperado que sob as condições dos parâmetros inicialmente utilizados nas funções, o modelo de aprendizagem individual iria convergir para o equilíbrio de Nash em que a saída média das firmas individualmente seria de 998.2. A resposta aqui encontrada, mostrou que as firmas não convergem de maneira clara para o equilíbrio de Nash, mas se aproximam desse equilíbrio. Outro fato também a ser considerado é que não são todas as 40 firmas do mercado que se aproximam desse equilíbrio, pois existem aquelas que tendem a se aproximar do equilíbrio Walrasiano.

Com a mudança para a nova função de demanda inversa para novos parâmetros das funções, a convergência dos valores das firmas para o equilíbrio Walrasiano foi clara, sendo que todas as firmas apresentaram o mesmo tipo de comportamento, não só algumas indo em direção a esse resultado.

Os resultados encontrados também contrapõem as colocações referentes em relação aos parâmetros que foram utilizados nas primeiras simulações, pois mesmo sob o uso desses valores os resultados não convergiram totalmente para o equilíbrio de Nash, com algumas firmas convergindo para o equilíbrio walrasiano.

Em relação aos resultados encontrados para o modelo de aprendizagem social, as respostas que eram esperadas de serem encontradas sob as condições das mesmas funções e parâmetros destas utilizados inicialmente, não se confirmaram. A modelagem mostrou que algumas firmas convergiram para o equilíbrio Walrasiano enquanto, outras convergiram para o modelo de Nash e algumas nem sequer convergiram para algum equilíbrio. Os resultados para as firmas em uma mesma simulação levaram a resultados inconclusivos sob a convergência de equilíbrios, pois os mesmos se mostram dispersos para as diferentes firmas.

Para que a análise continuasse, a função de demanda inversa utilizada para se encontrar o preço de equilíbrio e posteriormente os valores dos lucros foi alterada. A mudança feita levou

a resultados satisfatórios quanto ao equilíbrio obtido, pois sob a nova condição o equilíbrio Walrasiano foi claramente obtido em que todas as firmas convergiram para o equilíbrio.

Outra abordagem feita e que se mostrou influente nos resultados foram as simulações com variações nos números das firmas que compõem o mercado. Na aprendizagem individual com a presença de 10 e 20 firmas os resultados convergiram para o equilíbrio de Nash enquanto com 60 firmas houve a total convergência para o equilíbrio Walrasiano. Para o modelo de aprendizagem social, os resultados foram inconclusivos, sem equilíbrios específicos, para 10, 20 e 40 firmas enquanto para 60 firmas o equilíbrio Walrasiano foi encontrado. Esses resultados estão relacionados ao comportamento das firmas em relação as demais firmas, pois, com o aumento das firmas no mercado as mesmas compreendem que não conseguem influenciar o preço de mercado e se comportam como tomadoras de preço, conseqüentemente, indo em direção ao equilíbrio Walrasiano.

Mesmo diante das controvérsias apresentadas com os primeiros valores utilizados em relação aos modelos de aprendizagens social e individual, a MBA mostrou que os equilíbrios alcançados após a mudança feitas nas funções foram claros em relação ao equilíbrio Walrasiano.

O presente trabalho contribui com a linha de pesquisa sobre os equilíbrios oligopolistas de modo que buscou mostrar como os agentes (firmas) realmente tendem a se comportar em um mercado, já que foi usado da modelagem baseada em agentes e mostrou a racionalidade que os agentes podem ter de modo a buscarem melhores resultados para suas maximizações.

Muitos trabalhos podem ser originados para que o tipo de equilíbrio possa ser encontrado, como fazendo uso de outro algoritmo de aprendizagem ou outro método de seleção de melhores estratégias. Outra questão que pode ser abordada é a mudança do custo marginal que foi considerado constante nas simulações feitas na MBA.

Assim, as questões encontradas mostram que mesmo sem a presença do efeito de despeito, no qual as firmas tomam as ações que as prejudicam, mas que afetam mais aos outros do que elas mesmas, as firmas convergem para o equilíbrio Walrasiano sem que olhem para as estratégias dos concorrentes.

Referências

- ARIFOVIC, J.; MASCHEK, M. K. Revisiting individual evolutionary learning in the cobweb model—an illustration of the virtual spite-effect. *Computational economics*, Springer, v. 28, n. 4, p. 333–354, 2006.
- BENAROCH, M. Artificial intelligence in economics truth and dare. *Journal of Economic Dynamics and Control*, Elsevier, v. 20, n. 4, p. 601–605, 1996.
- BISCHI, G. I.; LAMANTIA, F.; RADI, D. An evolutionary cournot model with limited market knowledge. *Journal of Economic Behavior & Organization*, Elsevier, v. 116, p. 219–238, 2015.
- COPELAND, B. J. *The essential turing*. [S.l.]: Clarendon Press, 2004.
- DAWID, H.; KOPEL, M. On economic applications of the genetic algorithm: a model of the cobweb type. *Journal of Evolutionary Economics*, Springer, v. 8, n. 3, p. 297–315, 1998.
- EIBEN, A. E.; SMITH, J. E. et al. *Introduction to evolutionary computing*. [S.l.]: Springer, 2003.
- GOLDBERG, D. E. *Genetic algorithms*. [S.l.]: Pearson Education India, 2006.
- HOLLAND, J. H. *Adaptation in natural and artificial systems: an introductory analysis with applications to biology, control, and artificial intelligence*. [S.l.]: MIT press, 1992.
- MITCHELL, M. *An introduction to genetic algorithms*. [S.l.]: MIT press, 1998.
- NEUMANN, J. V. The general and logical theory of automata. *Cerebral mechanisms in behavior*, New York: John Wiley & Sons, v. 1, n. 41, p. 1–2, 1951.
- NICHOLSON, W.; SNYDER, C. *Microeconomic theory: Basic principles and extensions*. [S.l.]: Nelson Education, 2011.
- OSBORNE, M. J. *An introduction to game theory*. [S.l.]: Oxford university press New York, 2004.
- RAILSBACK, S. F.; GRIMM, V. *Agent-based and individual-based modeling: a practical introduction*. [S.l.]: Princeton university press, 2012.
- SMITH, J. M.; PRICE, G. R. The logic of animal conflict. *Nature*, Nature Publishing Group, v. 246, n. 5427, p. 15, 1973.
- VALLÉE, T.; YILDIZOĞLU, M. Convergence in the finite cournot oligopoly with social and individual learning. *Journal of Economic Behavior & Organization*, Elsevier, v. 72, n. 2, p. 670–690, 2009.
- VRIEND, N. J. An illustration of the essential difference between individual and social learning, and its consequences for computational analyses. *Journal of economic dynamics and control*, Elsevier, v. 24, n. 1, p. 1–19, 2000.
- YILDIZOĞLU, M. Competing r&d strategies in an evolutionary industry model. *Computational Economics*, Springer, v. 19, n. 1, p. 51–65, 2002.

APÊNDICE A – Aprendizagem Individual

Codificação do programa NetLogo para os resultados obtidos no modelo de Aprendizagem Individual.

```
globals [
  conjestpos ; conjunto de estratégias possíveis
  listaposcoe ; lista possíveis coeficientes
  médiaec ; média estratégia corrente (a que está sendo utilizada)
  médiame ; média melhor estratégia
  maxv ; máximo valor do espaço de busca
  minv ; mínimo valor espaço de busca
  nperíodos ; número de períodos para executar a
  estratégia
  teste
  teste-me
  tamanho-torneio
  a ; parâmetro da demanda
  b ; parâmetro da demanda
  c ; parâmetro da demanda
  kf ; custo fixo
  km ; custo marginal
  fitness ; lucro médio
]
```

```
breed [firmas firma]
```

```
firmas-own [
  ec ; estratégia corrente
  u ; vetor de valor de fitness
  v ; vetor dos resultados do crossover
  minimalista ; lista de estratégias
  conjval est ; conjunto de valores em que cada um
  está relacionado a uma estratégia
  gp ; geração prévia
  ng ; nova geração
]
```

```

eb ; espaço de busca
médi aconj val est
meconj val est ; mel hor estratégi a do conjunto de
val ores das estratégi as
mel horestratégi a
lucro
preço
cromossomo; vetor bi nári o das estratégi as
estratégi as; vetor lucro/fi tness de conjunto de
cromossomos
]

extensi ons [rnd]

to setup
ca
create-fi rmas 40 [
  setxy random-xcor random-ycor
  set size 1
  hi de-turtl e
]
let mi nhal ista1 n-values 11 [[i] -> i
let mi nhal ista2 sort-by > mi nhal ista1
set li staposcoe (map ^ n-values 11 [2] mi nhal ista2)
set conj estpos sum li staposcoe
ask fi rmas [
  ; criar vetor de valores de fi tness
  set u []
  ; criar vetor de conjunto de valores que cada fi tness
  está rel acionado a uma estratégi a
  set conj val est []
  ; o valor máxi mo do espaço de busca
  set maxv 2047
  ; valor mí ni mo do espaço de busca
  set mi nv 0
  set eb maxv - mi nv
  ; criar 40 estratégi as representadas pel as cordas bi nári as
  para cada fi rma
  set mi nhal ista n-values 40 [n-values 11 [random 2]]

```

```

; show minha lista
foreach minha lista [
  [estratégia] ->
  let k sum (map * estratégia listaposcoe) / conj estpos
  ; cria lista de valores que cada fitness está relacionado
  a minha lista de estratégias
  set conjval est lput (minv + k * eb) conjval est]]
reset-ticks
ask turtles [inicializarglobais]
end

```

```

to inicializarglobais
  set nperíodos 5000
  set tamanho-torneio 30
  set a  $-1 * 10^{(-97)}$ 
  set b  $1.5 * 10^{(95)}$ 
  set c -39.99999997
  set kf  $-4.097 * 10^{-94}$ 
  set km 0
  set preço []
  set lucro []
end

```

```

to go
  if ticks > nperíodos [stop]
  ask turtles [
    calculo-lucro]
  set fitness [mean lucro] of turtles
  ask turtles [
    set estratégias []
    set cromossomo []
    set estratégias sort-by > lucro
    ; show estratégias
    set estratégias sublist estratégias 0 30
    ; show estratégias
    foreach estratégias [
      [melhoresest] ->
      ; show melhoresest
      ; show posição melhoresest lucro

```

```

    set cromossomo lput item position mel hores est
    lucro minha lista cromossomo
  ]
  ; show cromossomo
nova-geração
set lucro []
]
tick
end

to calculo-lucro
  foreach conj val est [
    [quantidade] -> let x position quantidade conj val est
    let preço rodada a + b * (sum [item x conj val est]
    of firmas) ^ c
    set preço lput preço rodada preço
    let lucro rodada (preço rodada) * quantidade -
    (km + kf * quantidade)
    set lucro lput lucro rodada lucro]
end

to nova-geração
  set v []
  let crianças []
  let ulist (map + estratégias n-values
length(estratégias) [sqrt ( max(estratégias) ^ 2 )])
  repeat 20 [
    let hlist rnd:weighted-n-of-list 2 u-list [[w] -> w]
    let yposition random 2
    let pai1 item (position (item yposition hlist) u-list) cromossomo
    let pai2 item (position (item (1 - yposition) hlist)
    u-list) cromossomo
    ifelse (random-float 100.0 < 0.95)
    [set crianças crossover pai1 pai2
      set v sentence crianças v
    ]
    [set v lput pai1 v
      set v lput pai2 v]]
  mutar

```

```

set ng minha lista
set conj val est []
  foreach minha lista [
    [estratégia] ->
    let k sum (map * estratégia listaposcoe) / conj estpos
    ; cria lista de valores que cada fitness está relacionado
    a minha lista de estratégias
    set conj val est lput (minv + k * eb) conj val est]
set mel horestratégia item 0 conj val est
end

```

```

to-report crossover [estratégia1 estratégia2]
  let ponto-de-corte 1 + random (length estratégia1 - 1)
  report list (sentence (sublist estratégia1 0 ponto-de-corte)
    (sublist estratégia2 ponto-de-corte
    length estratégia2))
    (sentence (sublist estratégia2 0 ponto-de-corte)
    (sublist estratégia1 ponto-de-corte
    length estratégia1))
end

```

```

to mutar
  let h []
  let p []
  set p item 0 minha lista
  (foreach v [
    [estratégia] ->
    let k map [[i] -> ifelse-value (random-float
    100.0 < 0.001) [1 - i] [i]]; verificar probabilidade
    de mutação
    estratégia
    set h lput k h]
  )
  set minha lista h
end

```

```

to-report quan [p]
  let o mean [p] of turtles

```

```
  report o  
end
```

```
to-report fit [x]  
  let y mean [x] of turtles  
  report y  
end
```

APÊNDICE B – Aprendizagem Social

Codificação do programa NetLogo para os resultados obtidos no modelo de Aprendizagem Social.

```
global s [
  conj estpos ; conjunto de estratégias possíveis
  listaposcoe ; lista possíveis coeficientes
  médi aec ; média estratégia corrente (a que está sendo
  utilizada)
  médi ame ; média melhor estratégia
  maxv ; máximo valor do espaço de busca
  minv ; mínimo valor espaço de busca
  nperíodos ; número de períodos para executar a estratégia
  teste
  teste-me
  tamanho-torneio
  a ; parâmetro da demanda
  b ; parâmetro da demanda
  c ; parâmetro da demanda
  kf ; custo fixo
  km ; custo marginal
  fitness ; lucro médio
  dict
  strategi astot
  strategi astotfi nal
  lucrotot
]
```

```
breed [firmas firma]
```

```
firmas-own [
  ec ; estratégia corrente
  u ; vetor de valor de fitness
  v ; vetor dos resultados do crossover
  mi nhal ista ; lista de estratégias
```

```

conj val est ; conjunto de valores em que cada um
está relacionado a uma estratégia
gp ; geração prévia
ng ; nova geração
eb ; espaço de busca
médi aconj val est
meconj val est ; melhor estratégia do conjunto de
valores das estratégias
mel horestratégia
lucro
preço
cromossomo; vetor binário das estratégias
estratégias; vetor lucro/fitness de conjunto de cromossomos
]

```

```

extensões [rnd]

```

```

to setup
  ca
  create-firmas 40 [
    setxy random-xcor random-ycor
    set size 1
    hide-turtle
  ]
  let minhaLista1 n-values 11 [[i] -> i]
  let minhaLista2 sort-by > minhaLista1
  set listaposcoe (map ^ n-values 11 [2] minhaLista2)
  set conjestpos sum listaposcoe
  ask firmas [
    ; criar vetor de valores de fitness
    set u []
    ; criar vetor de conjunto de valores que
    cada fitness está relacionado a uma estratégia
    set conj val est []
    ; o valor máximo do espaço de busca
    set maxv 2047
    ; valor mínimo do espaço de busca
    set minv 0
    set eb maxv - minv
  ]

```



```

; criar 40 estratégias representadas pelas
cordas binárias para cada firma
set minhaLista n-values 40 [n-values 11 [random 2]]
; show minhaLista
foreach minhaLista [
  [estratégia] ->
  let k sum (map * estratégia listaposcoe) / conj estpos
; cria lista de valores que cada fitness está
relacionado a minhaLista de estratégias
set conjVal est lput (minv + k * eb) conjVal est]]
reset-ticks
ask turtles [inicializarglobais]
end

```

```

to inicializarglobais
set nperíodos 5000
set tamanho-torneio 30
set a -1 * 10 ^ (-97)
set b 1.5 * 10 ^ (95)
set c -39.99999997
set kf -4.097 * 10 ^ -94
set km 0
set lucro 0
set preço []
set lucro []
set lucrotot []
set estrategiastot []
set estrategiastotfinal []
end

```

```

to go
  if ticks > nperíodos [stop]
  ask turtles [
    calculo-lucro]
set fitness [mean lucro] of turtles
foreach sort turtles [[the-turtle] ->
  ask the-turtle [
    set estratégias []
    set cromossomo []

```

```
set estratégias sort-by > lucro
; show estratégias
set estratégias sublist estratégias 0 30
; show estratégias
foreach estratégias [
  [mel horeseset] -> ; show mel horeseset
  set lucrotot lput mel horeseset lucrotot
  ; show posição mel horeseset lucro
  set cromossomo lput item posição mel horeseset
  lucro minimalista cromossomo
  set estratégiasot lput item posição
  mel horeseset lucro minimalista
  estratégiasot
]
; ask turtle 0 [
  ; show max(lucrotot)
  ; show item 0 estratégiasot
  ; show estratégias
  ; show cromossomo
; ]
set lucrotot sort-by > lucrotot
  foreach lucrotot[[mel horeseset] -> set
  estratégiasotfinal lput item posição
  mel horeseset lucrotot estratégiasot estratégiasotfinal ]
; show item 0 estratégiasotfinal
set lucrotot sublist lucrotot 0 30
; show item 0 lucrotot
foreach lucrotot[[mel horeseset] -> set estratégiasotfinal
lput item posição mel horeseset lucrotot
estratégiasot estratégiasotfinal ]
set estratégiasotfinal sublist estratégiasotfinal 0 30
; show item 0 estratégiasotfinal
; show cromossomo
nova-geração
set lucro [ ]
]
]
tick
end
```

```

to calculo-lucro
  foreach conjval est [
    [quantidade] -> let x position quantidade conjval est
    let preço rodada a + b * (sum [item x conjval est]
of firmas) ^ c
    set preço lput preço rodada preço
    let lucro rodada (preço rodada) * quantidade
    - (km + kf * quantidade)
    set lucro lput lucro rodada lucro
  set lucrotot[]
  set estrategi astot[]
  set estrategi astotfinal []]
end

```

```

to nova-geração
  set v []
  let crianças []
  let ulist (map + lucrotot n-values length(lucrotot)
[sqrt ( max(lucrotot) ^ 2
)])
  repeat 20 [
    let hlist rnd:weighted-n-of-list 2 ulist [[w] -> w]
    let yposition random 2
    let pai1 item (position (item yposition hlist)
ulist) estrategi astotfinal
    let pai2 item (position (item (1 - yposition) hlist)
ulist) cromossomo
    ifelse (random-float 100.0 < 0.95)
    [set crianças crossover pai1 pai2
      set v sentence crianças v
    ]
    [set v lput pai1 v
      set v lput pai2 v]]
  mutar
  set ng mi nhal ista
  set conjval est []
  foreach mi nhal ista [

```

```

    [estratégia] ->
    let k sum (map * estratégia listaposcoe) / conj estpos
    ; cria lista de valores que cada fitness
    está relacionado a minha lista de estratégias
    set conj val est lput (minv + k * eb) conj val est]
  set mel horestratégia item 0 conj val est
end

to-report crossover [estratégia1 estratégia2]
  let ponto-de-corte 1 + random (length estratégia1 - 1)
  report list (sentence (sublist estratégia1 0 ponto-de-corte)
    (sublist estratégia2 ponto-de-corte length
    estratégia2))
    (sentence (sublist estratégia2 0 ponto-de-corte)
    (sublist estratégia1 ponto-de-corte length
    estratégia1))
end

to mutar
  let h []
  (foreach v [
    [estratégia] ->
    let k map [[i] -> ifelse-value
    (random-float 100.0 < 0.001) [1 - i] [i]]
    ; verificar probabilidade de mutação
    estratégia
    set h lput k h]
  )
  set minha lista h
end

to-report fit [x]
  let y mean [x] of turtles
  report y
end

```

